

# SWORD

XDI-SWORD-IMG-003

## Lab3: 像素处理实验 2: 对比度变换和亮度变换

Joseph Xu

2018-4-17

修改记录


版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-4-17

审核记录

姓名	职务	签字	日期

## 目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	6
1.1 概述.....	6
1.2 实验目标.....	7
1.3 实验条件.....	7
1.4 实验原理.....	7
2. PARTA: 对比度变换实验流程.....	11
2.1 操作步骤.....	11
3. PARTA: 对比度变换实验结果.....	27
4. PARTB: 亮度变换实验流程.....	28
3.1 操作步骤.....	28
5. PARTB 亮度变换实验结果.....	37

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	2 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

## 图目录

图 1-1	对比度变换.....	6
图 1-2	亮度变换.....	7
图 1-3	实验连接示意图.....	8
图 1-4	图像对比度调节连接示意图.....	8
图 1-5	ContrastTransform IP .....	8
图 1-6	LightnessTransform IP .....	10
图 2-1	复制一个实验 1 副本.....	11
图 2-2	重命名实验目录.....	11
图 2-3	启动 Vivado.....	12
图 2-4	打开工程.....	12
图 2-5	设置：添加 IP 库.....	13
图 2-6	选择图像 IP 库.....	13
图 2-7	图像处理 IP 库中的 IP 列表显示 .....	14
图 2-8	实验初始视图.....	15
图 2-9	添加 ContrastTransform IP .....	15
图 2-10	断开 DVI2RGB IP 的 RGB 端口 .....	16
图 2-11	断开端口后视图.....	16
图 2-12	展开 DVI2RGB IP 的 RGB 端口 .....	17
图 2-13	RGB 端口展开后视图.....	17
图 2-14	连接端口.....	18
图 2-15	添加 Constant IP .....	19
图 2-16	添加 Constant IP 后的视图 .....	19
图 2-17	Constant IP 设置 .....	20
图 2-18	连接 ContrastTransform IP 后的视图 .....	20
图 2-19	端口连接检查.....	21
图 2-20	保存设计.....	21
图 2-21	创建实验顶层 Wrapper 文件 .....	22
图 2-22	自动更新顶层文件.....	22
图 2-23	Generate Bitstream .....	22
图 2-24	点击 Yes 确认生成 bit 文件 .....	22
图 2-25	打开 Hardware Manager .....	23




	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	3 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

图 2-26	硬件连接对应位置.....	24
图 2-27	实际硬件连接.....	24
图 2-28	Open target .....	25
图 2-29	Program Device .....	25
图 2-30	烧写目标器件.....	26
图 2-31	编程进度条.....	26
图 3-1	对比度变换显示结果.....	27
图 4-1	回到 Vivado 主界面继续添加 IP .....	28
图 4-2	添加亮度变换 IP : LightnessTransform .....	28
图 4-3	断开 RGB2DVI IP 的 vid_pVDE 端口 .....	29
图 4-4	删除 IP 后的视图.....	29
图 4-5	LightnessTransform IP 配置 .....	30
图 4-6	Constant IP 设置 .....	30
图 4-7	端口连接检查.....	31
图 4-8	保存设计.....	31
图 4-9	创建顶层文件.....	31
图 4-10	自动更新顶层文件.....	32
图 4-11	Generate Bitstream.....	32
图 4-12	点击 Yes 确认生成 bit 文件 .....	32
图 4-13	打开 Hardware Manager .....	33
图 4-14	硬件连接对应位置.....	34
图 4-15	实际硬件连接.....	34
图 4-16	Open target .....	35
图 4-17	Program Device .....	35
图 4-18	烧写目标器件.....	36
图 4-19	编程进度条.....	36
图 5-1	亮度变换显示结果.....	37

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	4 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

## 表目录

表 1	DVI2RGB IP 端口列表 .....	9
表 2	RGB2DVI IP 端口列表 .....	10

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	5 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

## 1. 实验简介

该实验以实验 2 为基础，其中对比度变换为将灰度化后的像素按照设定的对比度进行重新计算，从而得到更细腻的颜色表达；亮度变换部分为设定亮度增益后，对每个像素数据都附加该增益后得到新像素值。

- **对于初学者，整个实验预计耗时 1 小时。**
- **对于进阶者，整个实验预计耗时 10 分钟。**

### 1.1 概述

对比度指的是一幅图像中明暗区域最亮的白和最暗的黑之间不同亮度层级的测量，差异范围越大代表对比越大，差异范围越小代表对比越小。对比度变换属于图像增强的一种，图像增强，即增强图像中有用的信息，其目的是改变图像的视觉效果，针对应用刻意强调图像整体或局部特征，是一个失真的过程。对比度变换是最基础的图像增强运算之一，由于人眼不仅仅是根据色彩的绝对值，还会根据某个区域和其周边的一个对比来得到整体的感受，这个“对比”量化后即为对比度。对比度变换的方式有很多，但差异基本都是变换系数所造成的，系数为常数的变换为线性变换，否则为非线性变换，本实验所使用的对比度变换方法为线性变换。

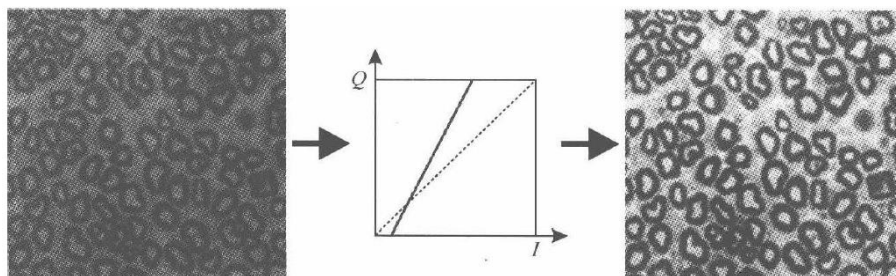



图 1-1 对比度变换

亮度变换同样是最基础的图像增强运算之一，亮度实际上是像素的各个通道色彩分量的一个线性函数，故可以通过更改每个通道的色彩值来进行亮度的变换。所以，亮度变换就是对图像中每一个像素的色彩进行增加或者减少的线性变换，与对比度变换相同，亮度变换的方式也有很多，差异也基本都是变换系数所造成的，系数为常数的变换为线性变换，否则为非线性变换，本节将讨论如何实现线性的亮度变换。

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	6 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

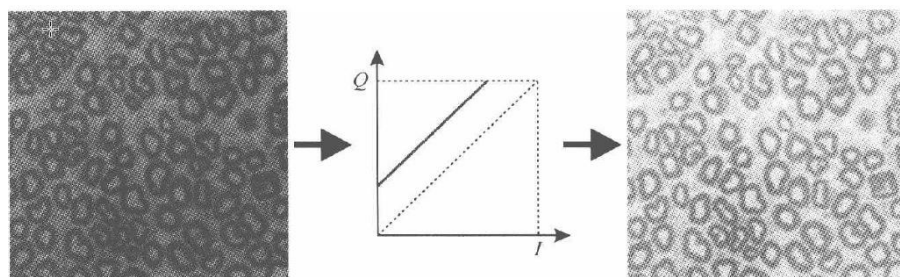


图 1-2 亮度变换

## 1.2 实验目标

本实验的目标为 SWORD4.0 能够正常地在 HDMI 显示器上输出视频画面，其中：

1. 1280x720@60Hz——对比度变换视频画面；
2. 1280x720@60Hz——亮度变换视频画面；


## 1.3 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式计算机 HDMI 输出/带 HDMI 输出的视频机顶盒
	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
软件	Vivado Design Suite	1	版本：2014.4
	视频接口 IP 库	1	FPGA-Image-Library.zip*

\*注：FPGA-Image-Library 为戴天宇开发的一个开源图像处理 IP 库，该 IP 库遵循 LGPL，详情请见：<http://fil.dtysky.moe>

## 1.4 实验原理

该实验的连接方式如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	7 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		



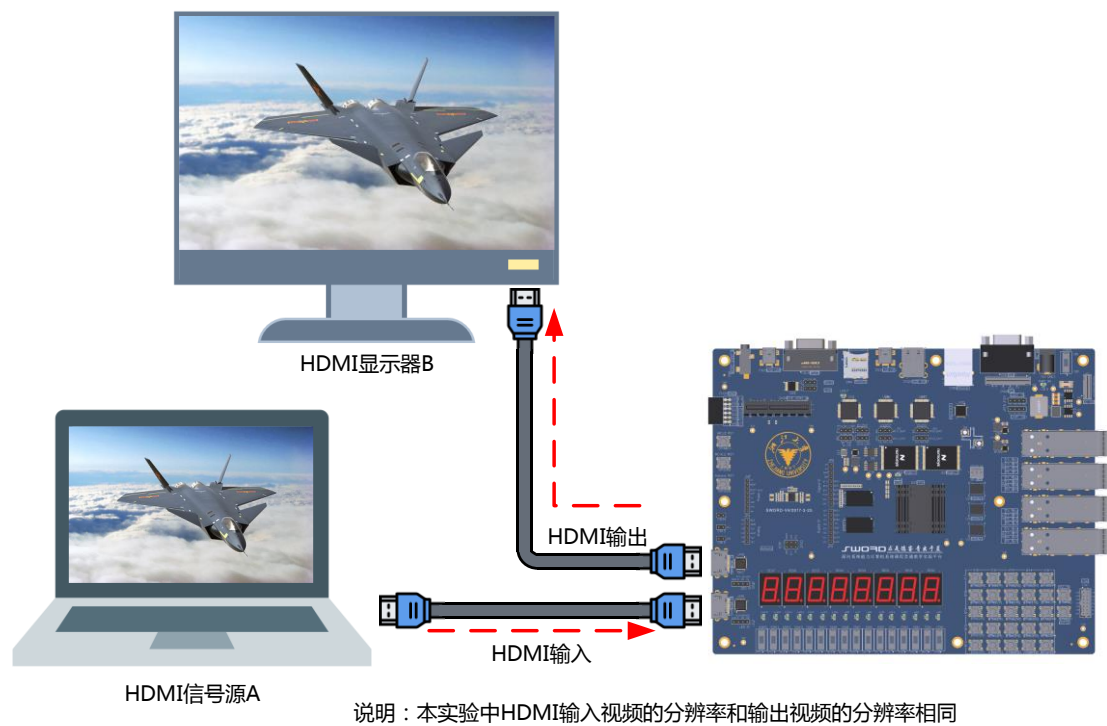


图 1-3 实验连接示意图

实验利用了 1 个 IP 来实现对比度变换：ContrastTransform。其中：对比度变换实验 IP 连接示意图如下图所示：

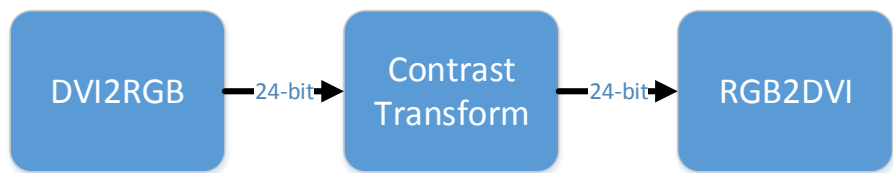


图 1-4 图像对比度调节连接示意图

亮度变换实验 IP 连接示意图如下图所示：

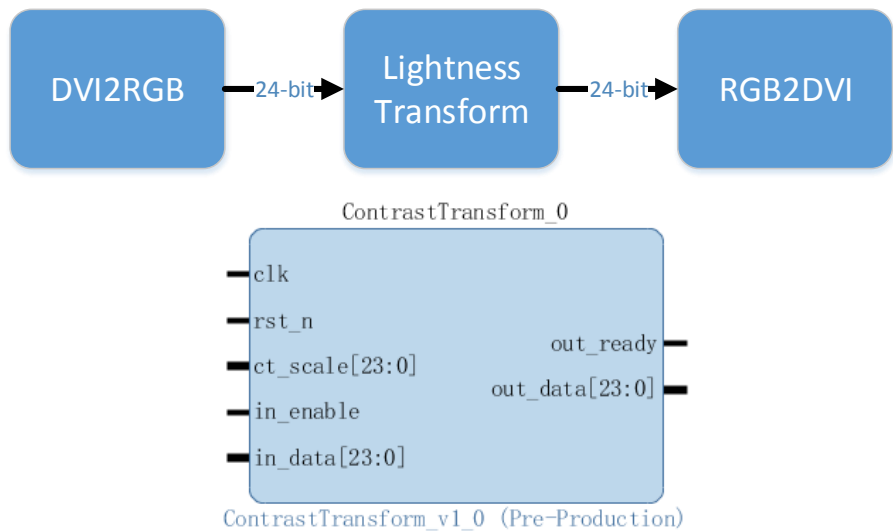


图 1-5 ContrastTransform IP


	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	8 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

该 IP 的端口信号定义如下表所示：

表 1 DVI2RGB IP 端口列表

信号名	方向	宽度	含义
RefClk	I	1	200 MHz reference clock.
aRst_n	I	1	<b>Asynchronous reset of configurable polarity. Assert, if RefClk is not within spec.</b>
pRst_n	I	1	Active-high reset synchronous with PixelClk. Configurable polarity.
TMDS_Clk_p	I	1	TMDS Clock Channel.
TMDS_Clk_n	I	1	TMDS Clock Channel.
TMDS_Data_p	I	3	TMDS Data Channel 0.
TMDS_Data_n	I	3	TMDS Data Channel 0.
DDC_SCL	O	1	Display Data Channel clock signal. Optional.
DDC_SDA	IO	1	Display Data Channel data signal. Optional.
PixelClk	O	1	Pixel clock recovered from the TMDS clock channel. Driven by BUFR.
pVDE	O	1	Video data valid: • 1 = Active video. • 0 = Blanking period.
pHSync	O	1	Horizontal synchronization video timing signal.
pVSync	O	1	Vertical synchronization video timing signal.
pData	O	24	Video pixel data packed as RGB.
aPixelClkLckd	O		Active-high asynchronous locked signal for PixelClk. When low, PixelClk is lost or not within specs.
SerialClk	O		Fast clock, toggling at five times the frequency of PixelClk. It is used internally for deserialization.  Provided optionally as output for advanced use, like clocking an RGB2DVI core. Driven by BUFIO.

而 RGB2DVI 这个 IP 的作用是将像素数据按照特殊编码的方式转换成 TMDS 信号。

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	9 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

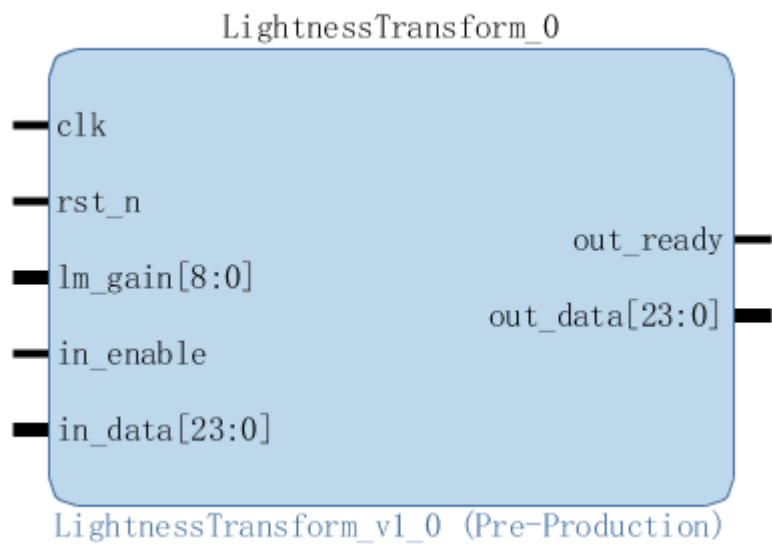


图 1-6 LightnessTransform IP

该 IP 的端口信号定义如下表所示：

表 2 RGB2DVI IP 端口列表

信号名	方向	宽度	含义
SerialClk	I	1	Optional fast serial clock. Has five times the frequency of PixelClk and are phase-aligned. Can be generated internally.
aRst(_n)	I	1	Asynchronous reset of configurable polarity. Assert, if PixelClk and SerialClk are not within spec.
Clk_p	O	1	DVI Clock Channel.
Clk_n	O	1	
Data_p	O	3	DVI Data Channel 0.
Data_n	O	3	
PixelClk	I	1	Pixel clock.
pVDE	I	1	Video data valid: <ul style="list-style-type: none"><li>• 1 = Active video.</li><li>• 0 = Blanking period.</li></ul>
pHSync	I	1	Horizontal synchronization video timing signal.
pVSync	I	1	Vertical synchronization video timing signal.
pData	I	24	Video pixel data packed as RBG.

## 2. PARTA: 对比度变换实验流程

本章将详细描述如何在 Vivado 2014.4 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。

### 2.1 操作步骤

- 1 由于本实验是在实验 1 的基础上进行扩展，所以我们将之前的实验部分复制 1 份，具体做法为在 D:\ImageLabs 文件夹下，将鼠标左键选中 lab1，然后按住 Ctrl 键不放，并拖拽到空白处，这样得到一个 lab1 的副本，如下图所示：

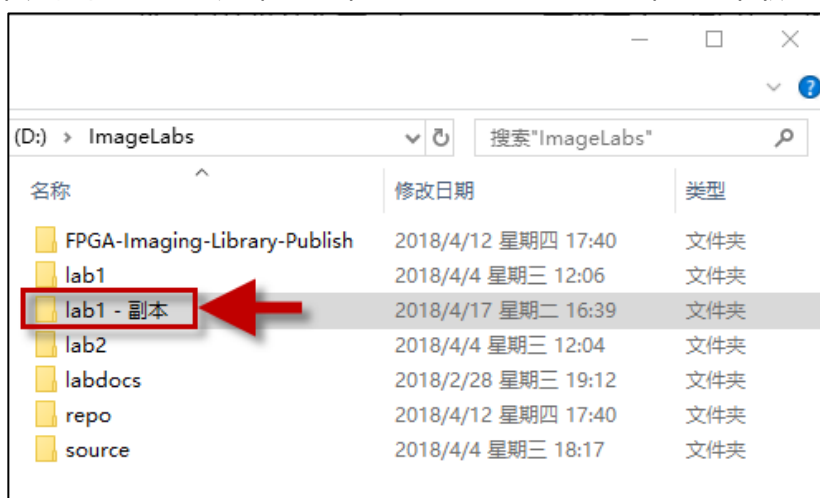


图 2-1 复制一个实验 1 副本

然后将 lab1 的副本重命名为 lab3，如下图所示，至此我们就可以在 lab3 文件夹里开始我们的实验内容：

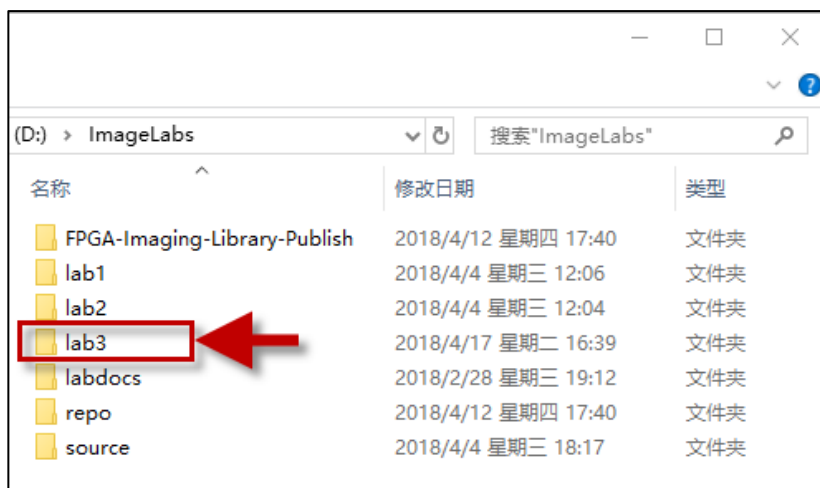



图 2-2 重命名实验目录

- 2 接着启动 Vivado 2014.4，在启动界面选择 Open Project，如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	11 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

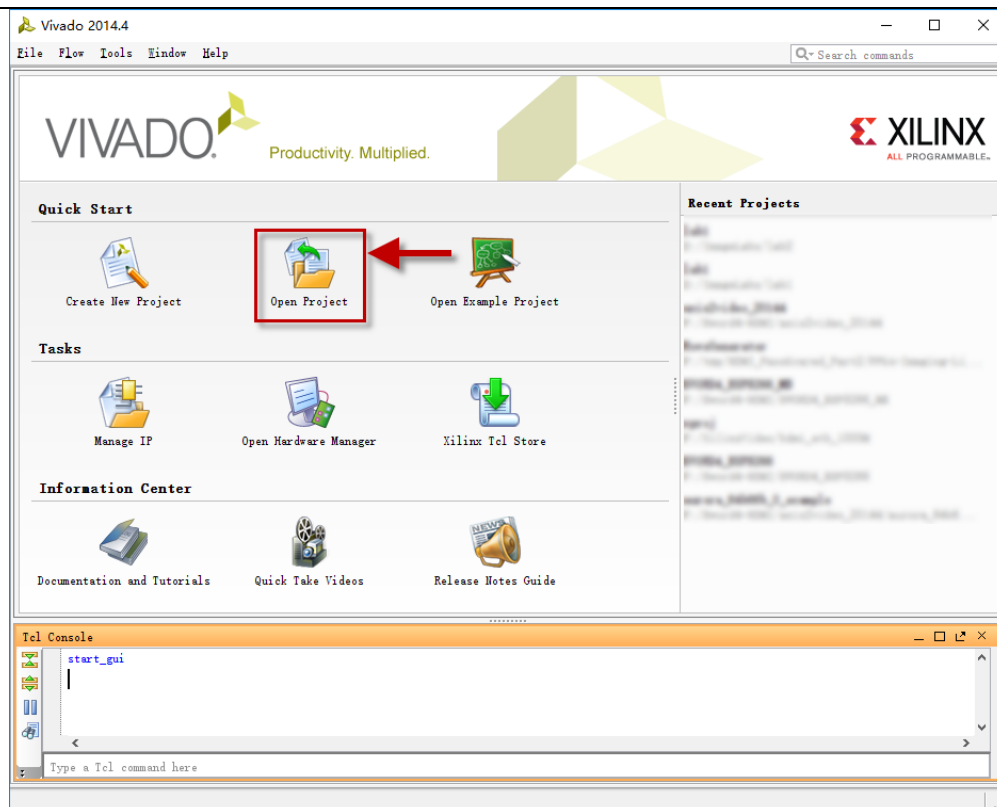


图 2-3 启动 Vivado

- 3 然后在选择对话框中 找到之前的 lab3 (即 D:\ImageLabs\lab3) 然后选择 lab1.xpr 文件, 点击 OK, 打开工程, 整个过程如下图所示:

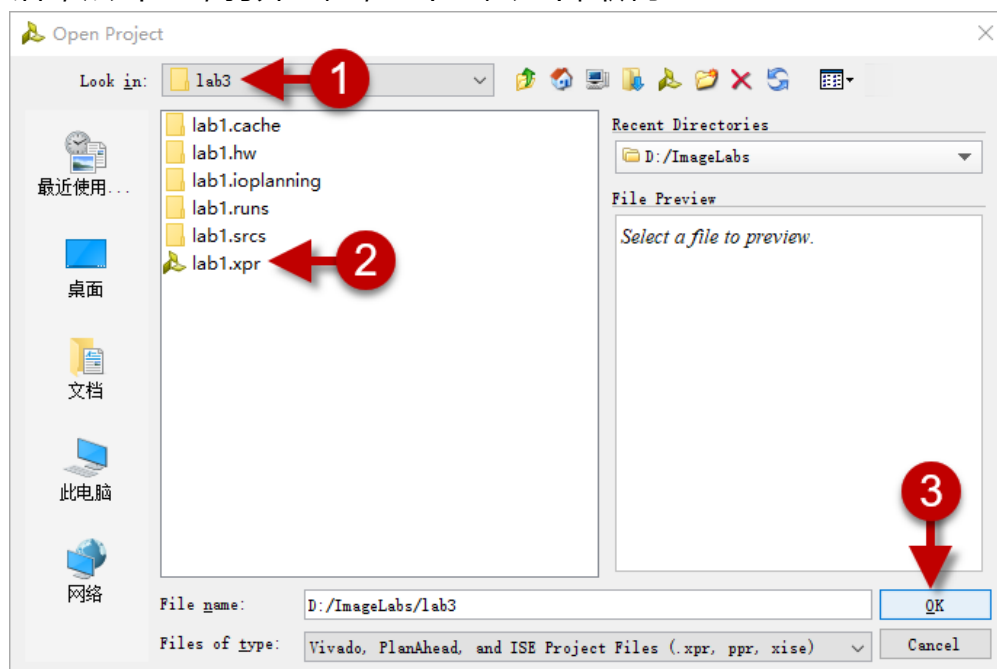



图 2-4 打开工程

- 4 接着我们将 FPGA-Image-Library IP 库添加进来, 在 Vivado 的主界面点击 Project Settings, 接着在新开对话框中点击 IP, 点击 Add Repository, 过程如下图所示:

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	12 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

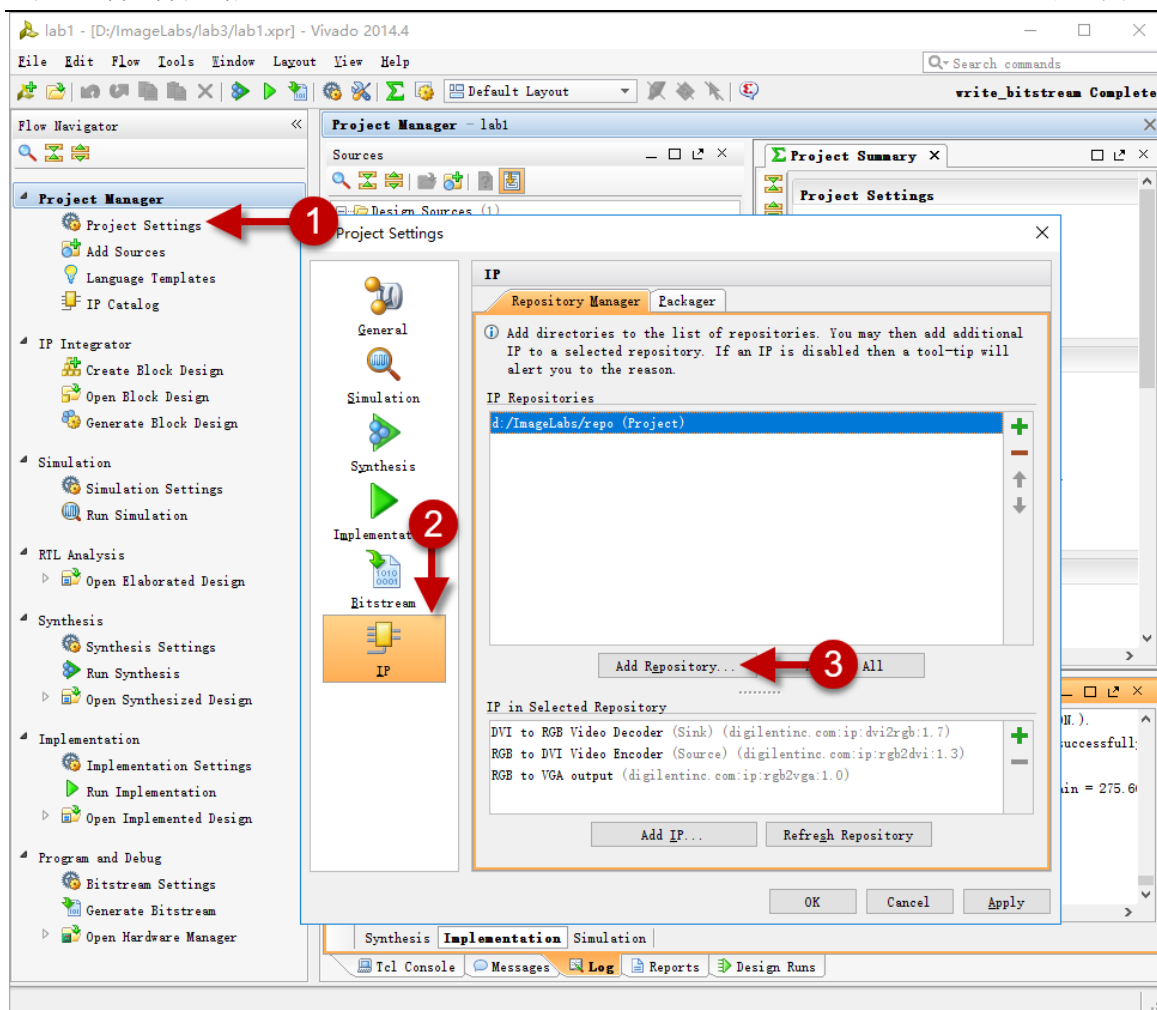


图 2-5 设置：添加 IP 库

- 5 在对话框中找到 D:\ImageLabs\FPGA-Image-Library-Publish，然后点击 Select，如下图所示：

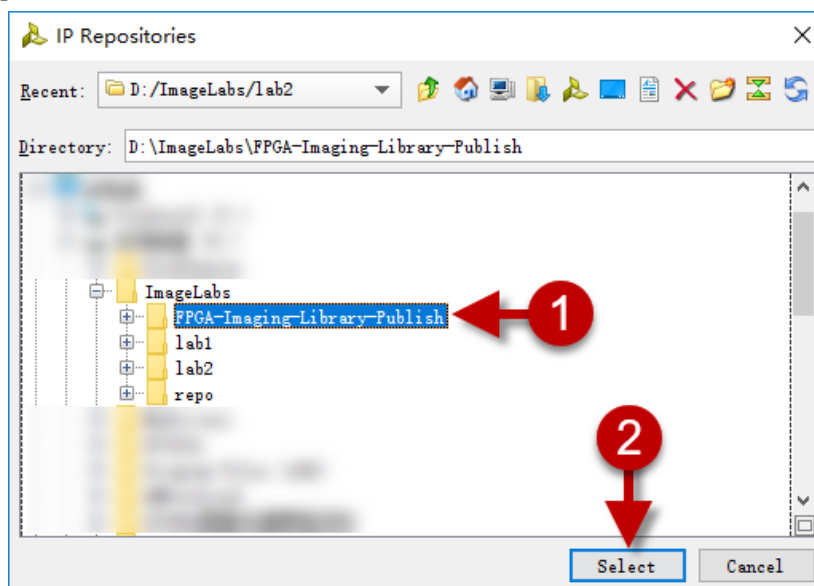



图 2-6 选择图像 IP 库

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	13 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

添加好 IP 库后，能看到 Vivado 会自动扫描库中的 IP，如果能看到如下图所示的一些 IP，则表示 IP 库添加成功，此时点击 OK 继续：

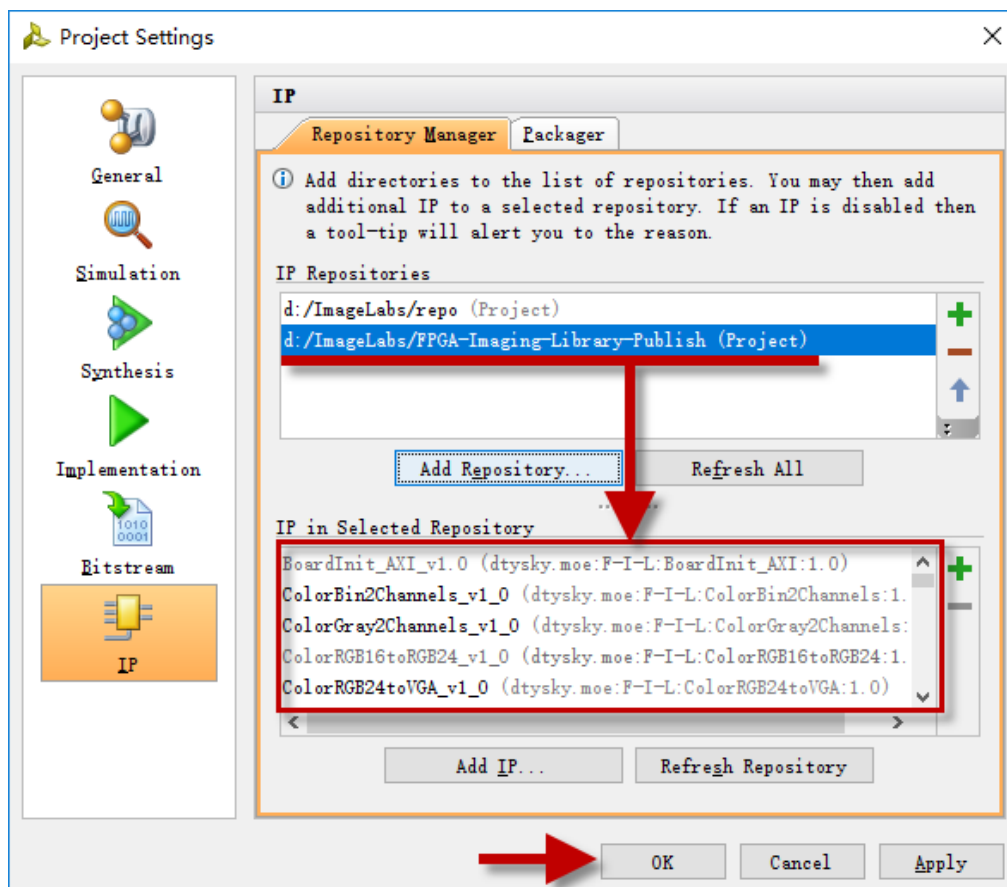



图 2-7 图像处理 IP 库中的 IP 列表显示

- 6 回到 Vivado 的主界面后，点击 Open Block Design，这时会在主界面右边区域看到之前实验 1 的 IP 结构，如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	14 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

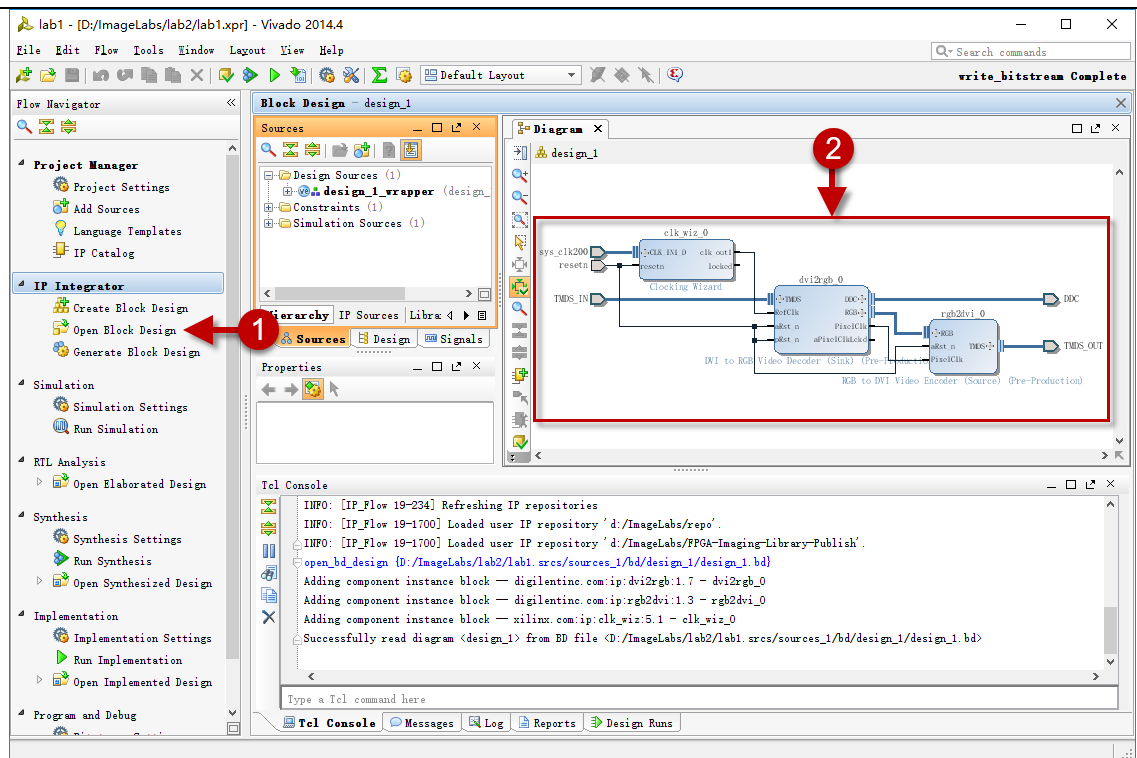


图 2-8 实验初始视图

- 7 在此基础上，我们开始添加 IP，点击左边栏的 Add IP 图标，然后在弹出的搜索框中，输入 Contrast，这时能看到搜索结果中有个 ContrastTransform 的 IP，双击它进行添加，整个过程如下图所示：

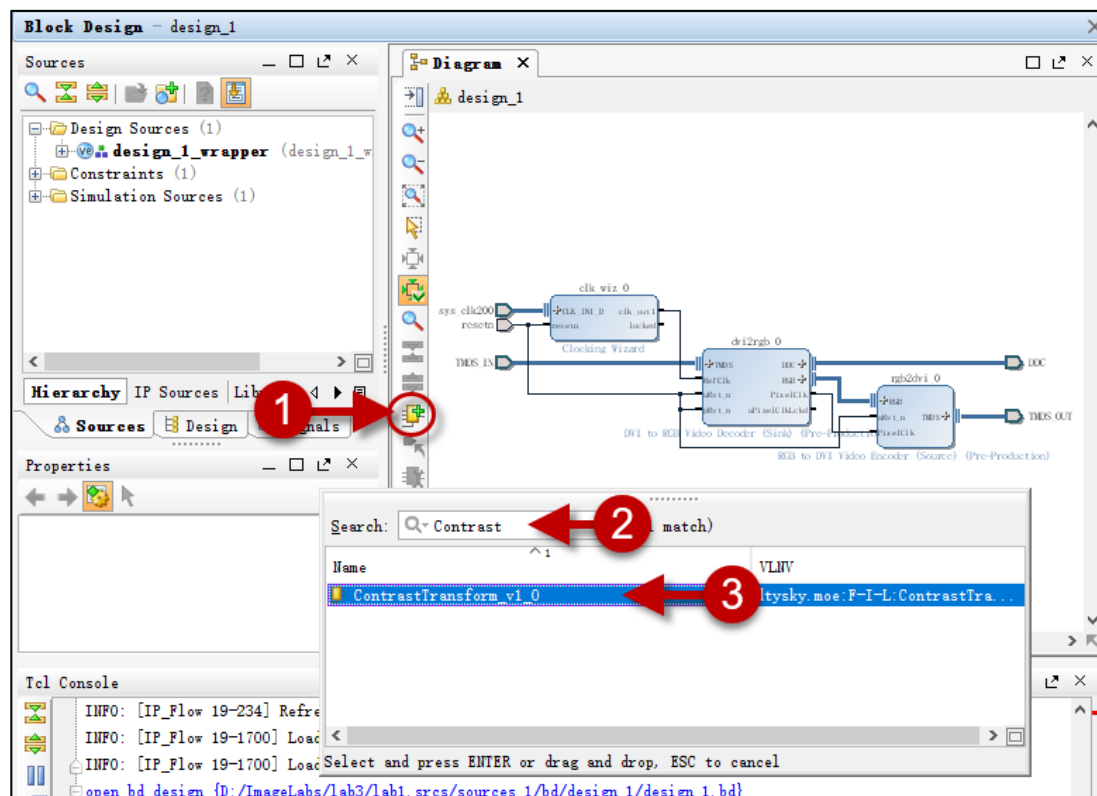



图 2-9 添加 ContrastTransform IP

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	15 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		



- 8 由于我们要将对比度变换 IP 接入到图像数据流中，因此在模块化设计视图中，我们需要现将之前的数据流先断开，为此我们先用鼠标左键选中 DVI2RGB IP 的 RGB 端口，此时会看到 RGB 信号高亮为浅黄色（**注意一定不要选中整个 IP**），然后鼠标右键单击，在弹出菜单中选择 Disconnect Pin，整个过程如下图所示：

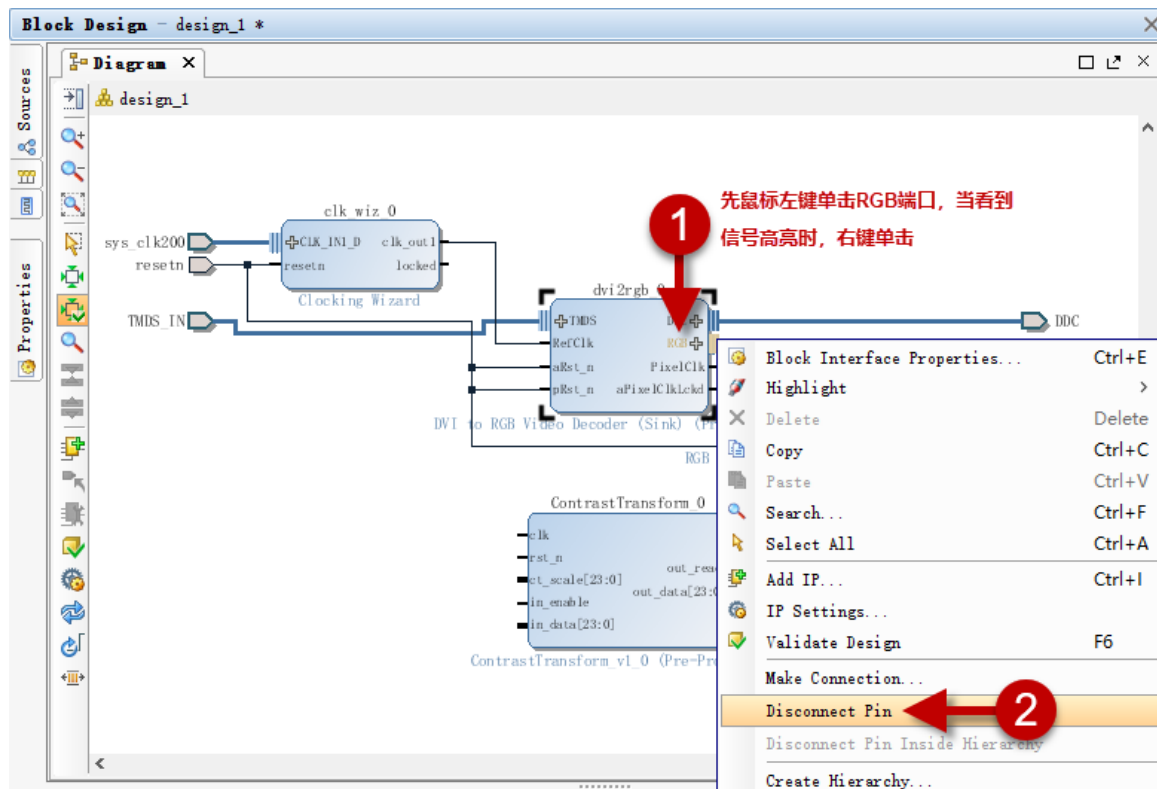


图 2-10 断开 DVI2RGB IP 的 RGB 端口

图像数据流断开后的 IP 视图如下图所示：

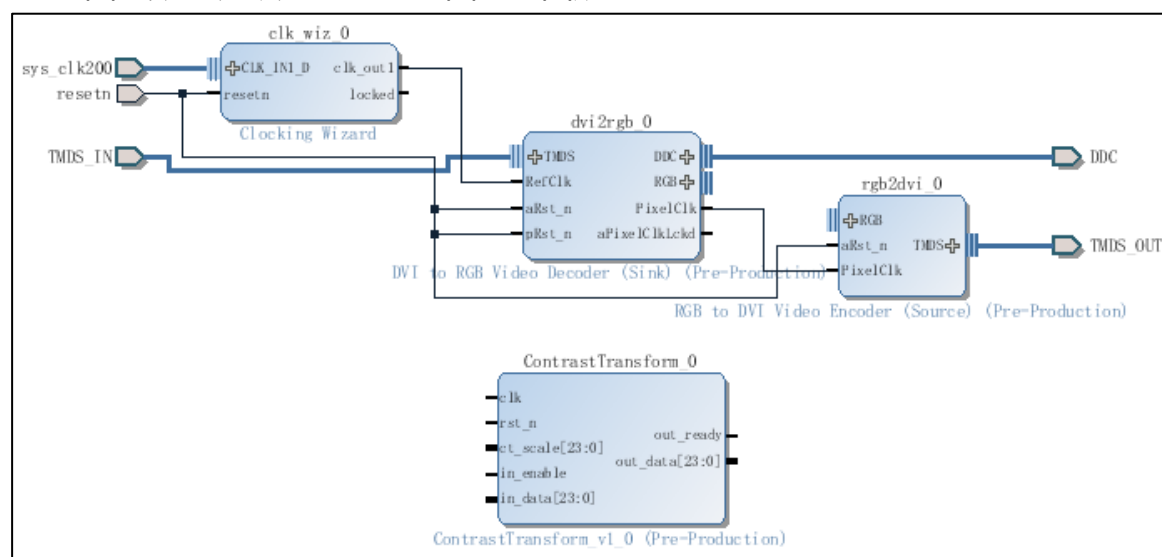




图 2-11 断开端口后视图

- 9 接着我们开始将灰度化 IP 接入，现将 DVI2RGB IP 的 RGB 端口展开，将鼠标移动到 DVI2RGB 的 RGB 端口，此时鼠标指针会变成一个  的符号，如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	16 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

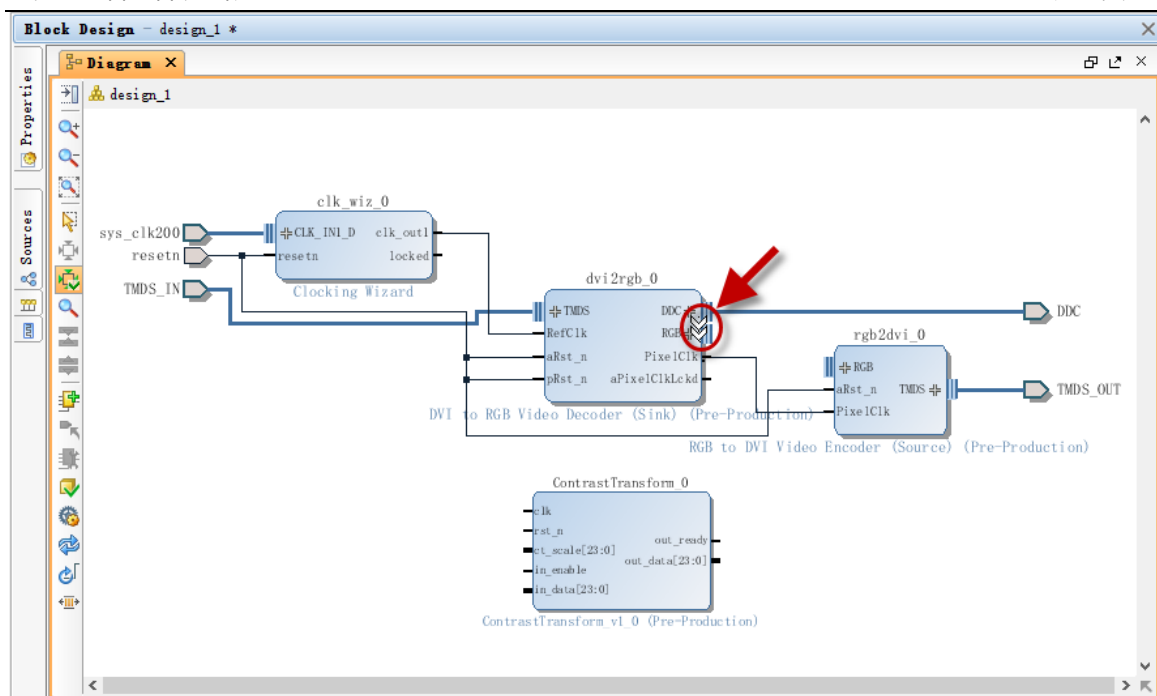


图 2-12 展开 DVI2RGB IP 的 RGB 端口

左键点击即可展开，如下图所示：

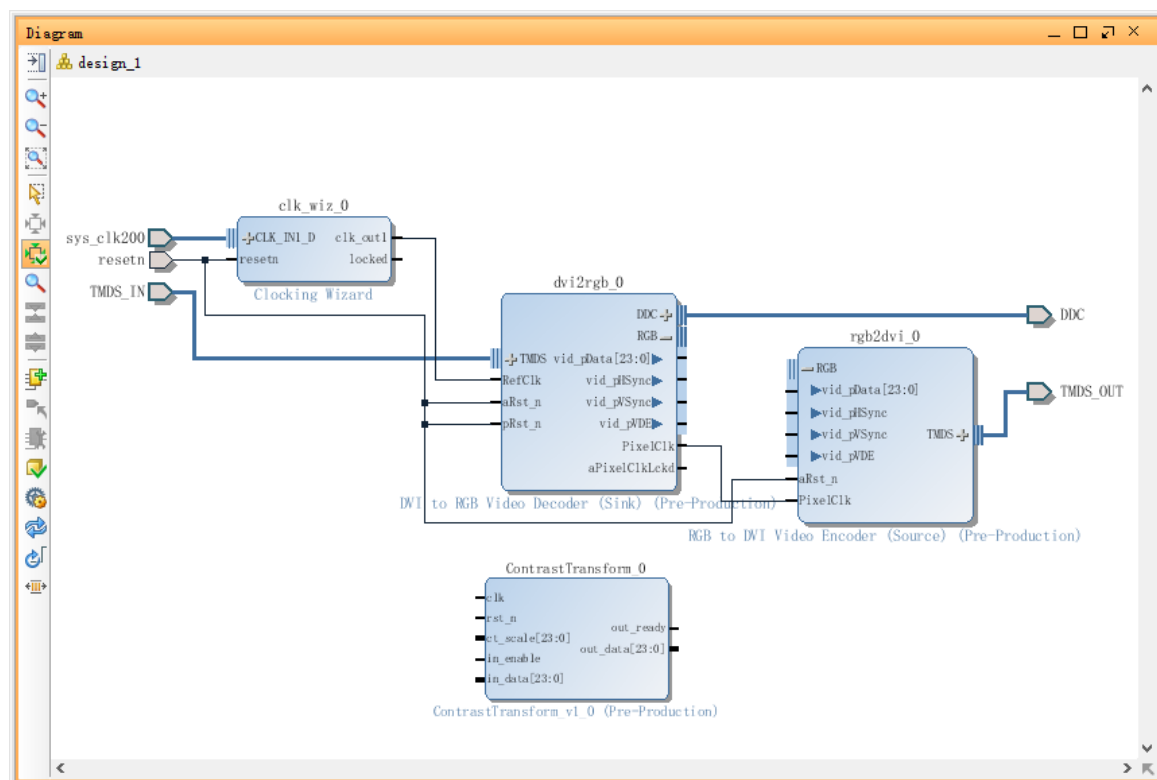



图 2-13 RGB 端口展开后视图

我们将按照如下方式先对 ContrastTransform IP 进行部分连接：

ContrastTransform\_0:clk → dvi2rgb\_0:PixelClk

ContrastTransform\_0:rst\_n → clk\_wiz\_0:resetn

ContrastTransform\_0:in\_enable → dvi2rgb\_0:vid\_pVDE

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	17 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

ContrastTransform\_0.in\_data[23:0] → dvi2rgb\_0.vid\_pData[23:0]

连接的操作如下图所示，鼠标光标移动到对应端口，然后点击不放并拖拽到与之对应的连接端口上，如果符合连接规则，端口会自动显示绿色的勾。

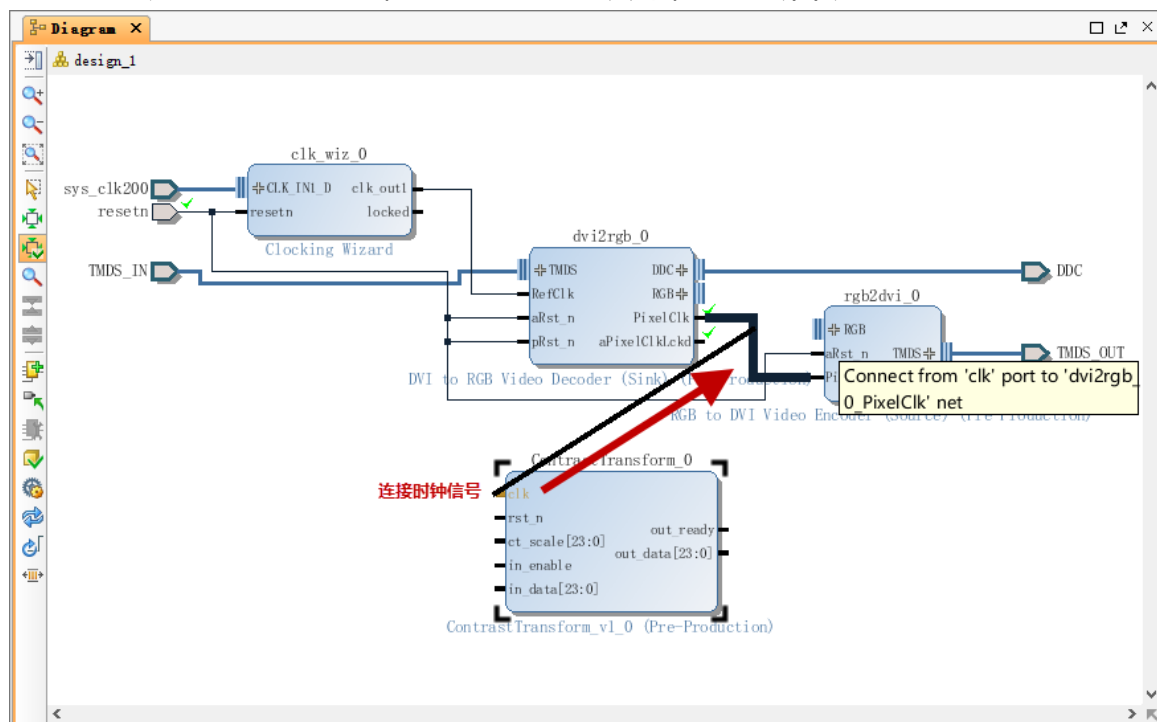



图 2-14 连接端口

- 10 由于对比度变换 IP 需要一个 24 位宽的线性变换系数，其中高 12 位为整数位，低 12 位为小数位。所以在本实验中，我们要设置一个常数作为该变换系数输入，我们直接调用一个 constant 的 IP 来实现输入，点击 Add IP 图标，在弹出的搜索栏中输入 constant，然后在结果栏中双击 Constant 这个 IP 进行添加，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	18 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

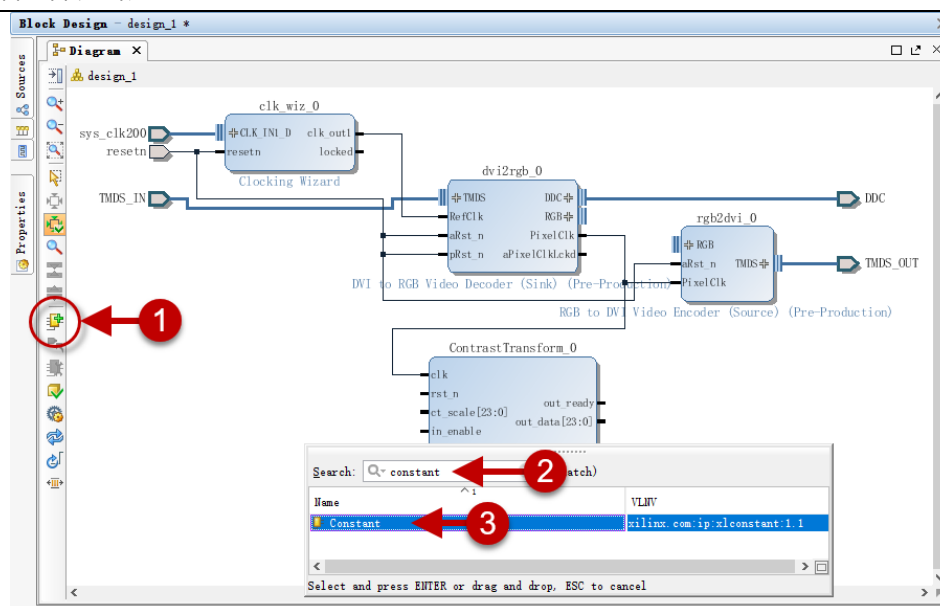


图 2-15 添加 Constant IP

添加后的 IP 视图如下图所示：

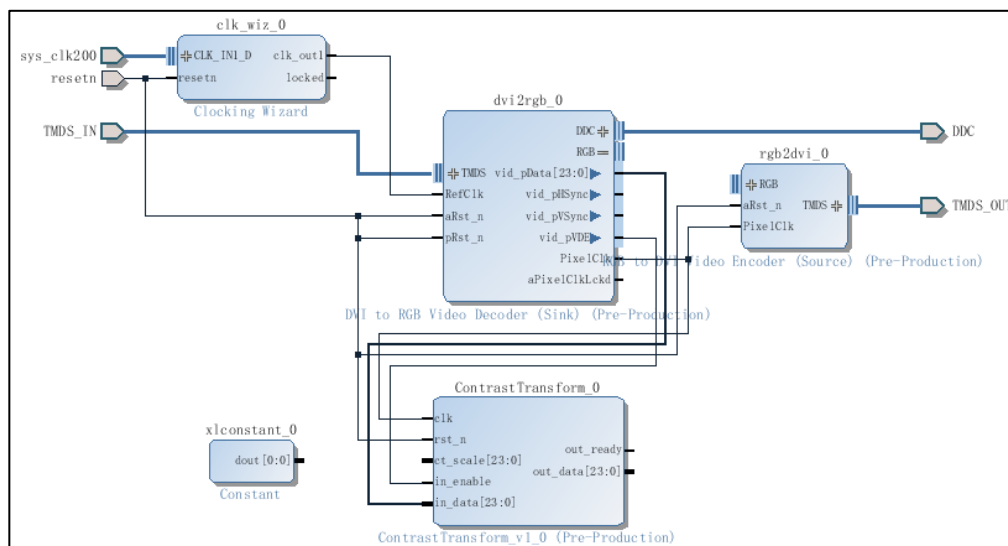


图 2-16 添加 Constant IP 后的视图

接着双击 xlconstant\_0 这个 IP 进行配置，在配置对话框中，进行如下设置：

Const Width: 3

Const Val: 13506 (注：等效于二进制0011.010011000010，即系数为 3.3)

完成设置后，点击 OK 确认，整个拖成如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	19 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

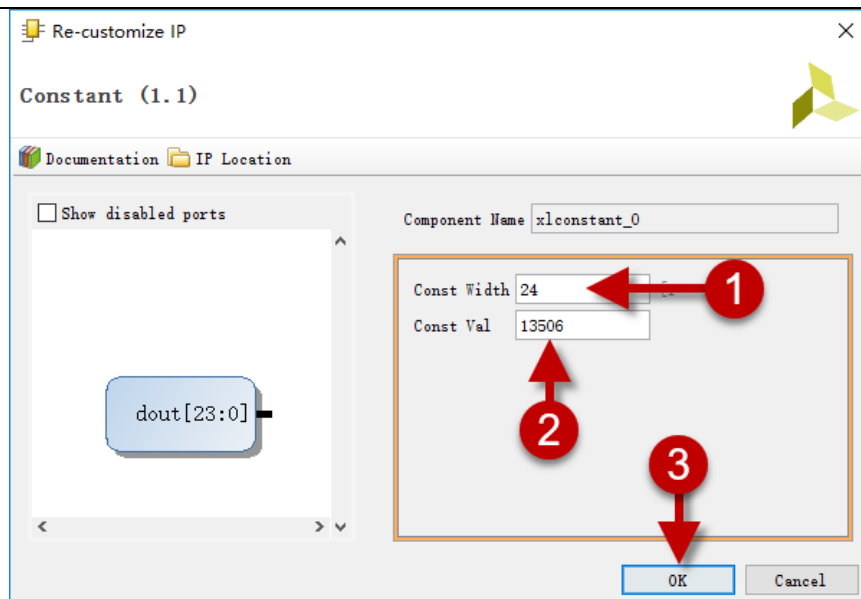


图 2-17 Constant IP 设置

- 11 之后将 ContrastTransform\_0 和 xlconstant\_0 这两个 IP 进行连接 连接方式如下：

ContrastTransform\_0:out\_ready → rgb2dvi\_0: vid\_pVDE

ContrastTransform\_0:out\_data[23:0] → rgb2dvi\_0: vid\_pData[23:0]

xlconstant\_0:dout[23:0] → ContrastTransform\_0:ct\_scale[23:0]

dvi2rgb\_0:vid\_pHSync → rgb2dvi\_0:vid\_pHSync

dvi2rgb\_0:vid\_pVSync → rgb2dvi\_0:vid\_pVSync

连接好后 IP 视图如下图所示：

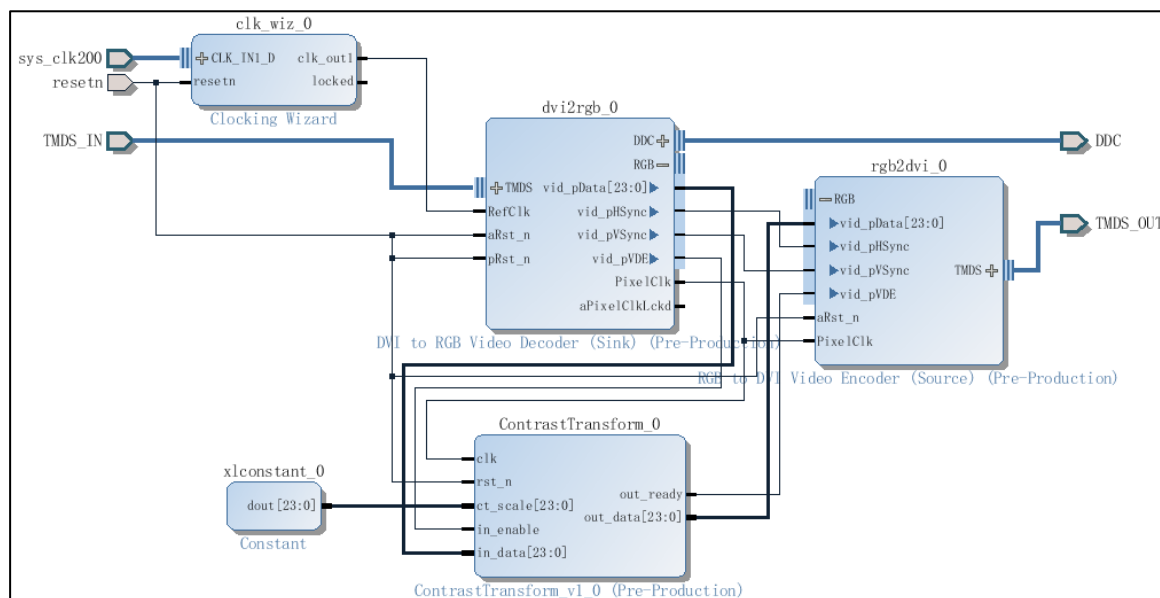


图 2-18 连接 ContrastTransform IP 后的视图

- 12 请仔细检查各 IP 的端口连接是否正确,为了方便核对,本文特别准备了各种连接的高亮色图以示区别,如下图所示：

**提示：下图仅作为检查连接使用，读者完全不必也按照图示颜色进行标注 !!!**

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	20 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

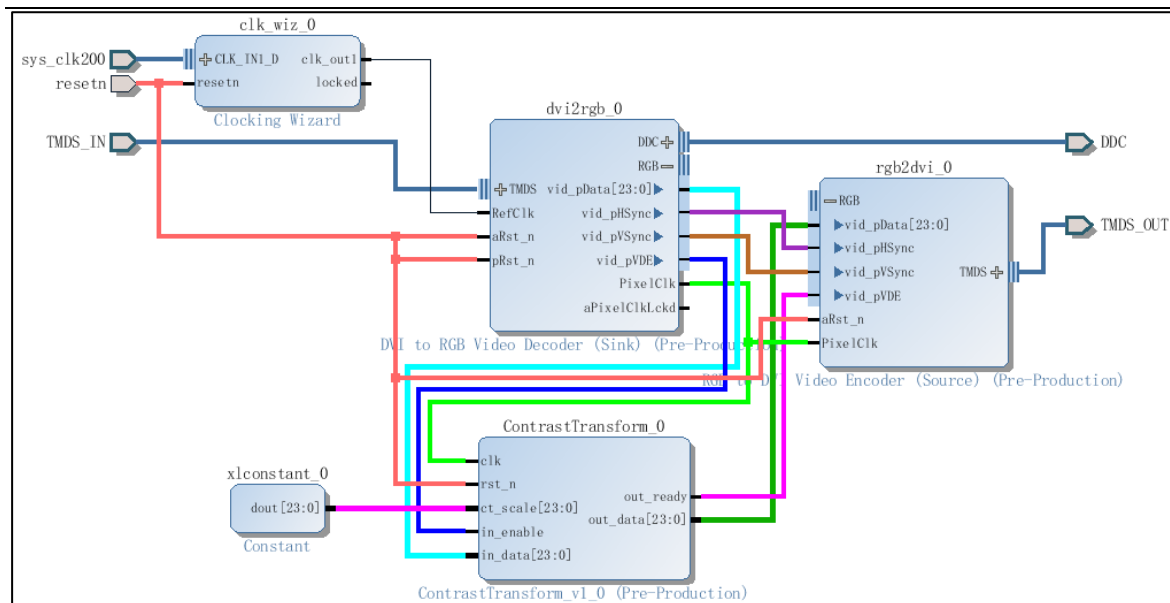


图 2-19 端口连接检查

- 13 连接检查无误后，即可保存 IP 模块化设计，在 Vivado 主界面点击保存图标，如下图所示：

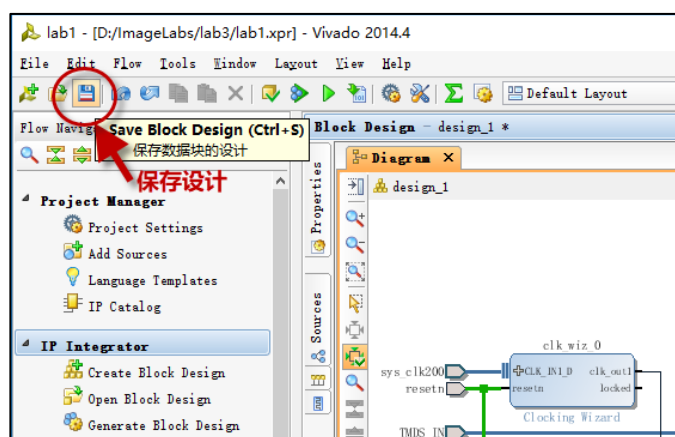
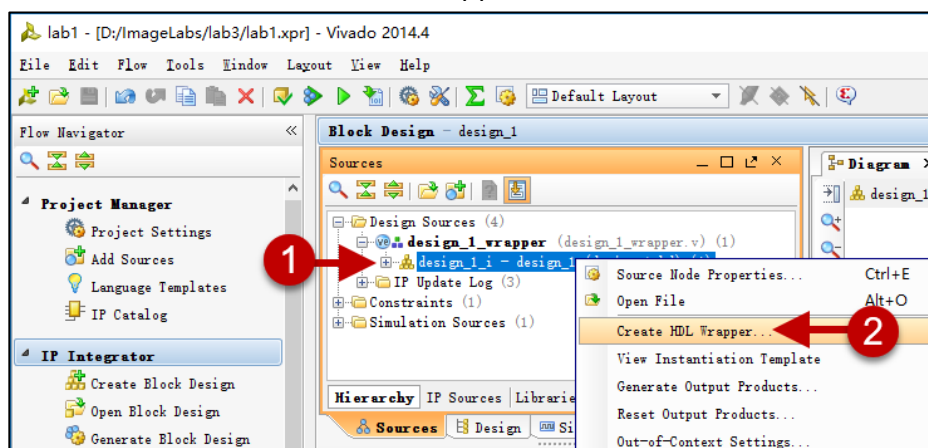


图 2-20 保存设计

接着在 Source 子窗口中展开 design\_1\_wrapper，选中 design\_1.bd，鼠标右键单击，在弹出的菜单中选择 Create HDL Wrapper，整个过程如下图所示：



	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	21 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

图 2-21 创建实验顶层 Wrapper 文件

接着在弹出的对话框中，保持默认选项不变，即选择 Let Vivado manage wrapper and auto-update，然后点击 OK，如下图所示：

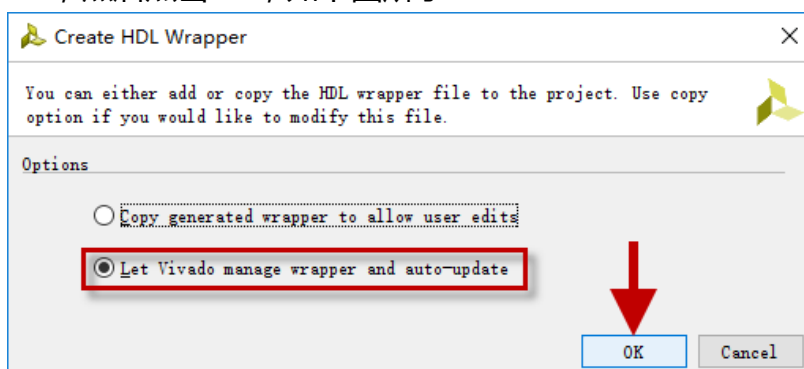


图 2-22 自动更新顶层文件

在 Vivado 主界面点击 Generate Bitstream，生成 bit 文件，如下图所示：

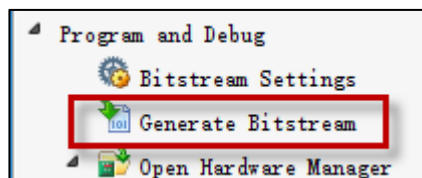


图 2-23 Generate Bitstream

在弹出的提示框中直接点 Yes 确认并继续，如下图所示：

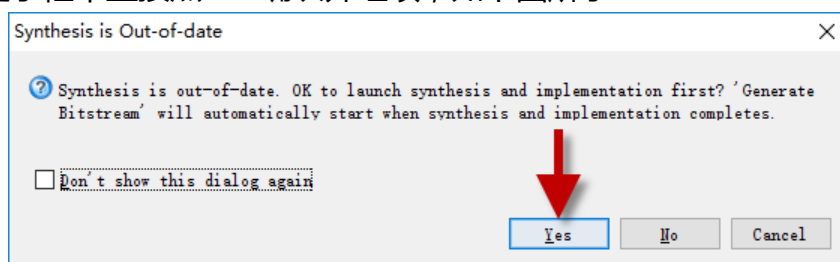



图 2-24 点击 Yes 确认生成 bit 文件

大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	22 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

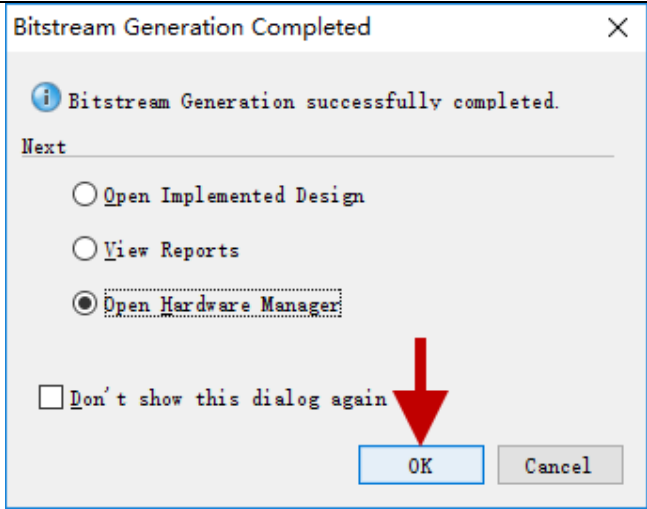


图 2-25 打开 Hardware Manager

接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	23 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		



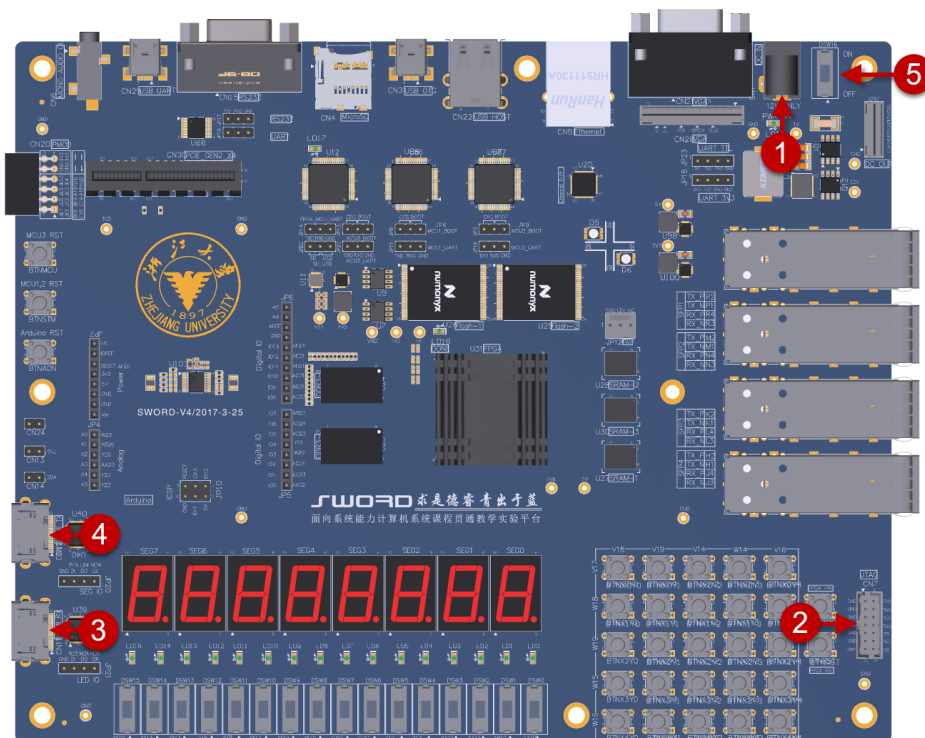



图 2-26 硬件连接对应位置

连接好后的效果如下图所示：



图 2-27 实际硬件连接

- 14 接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	24 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

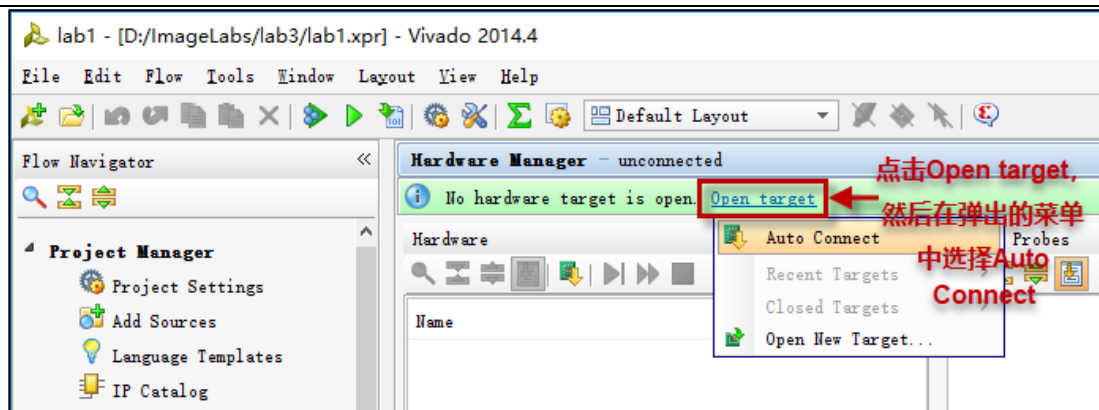


图 2-28 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

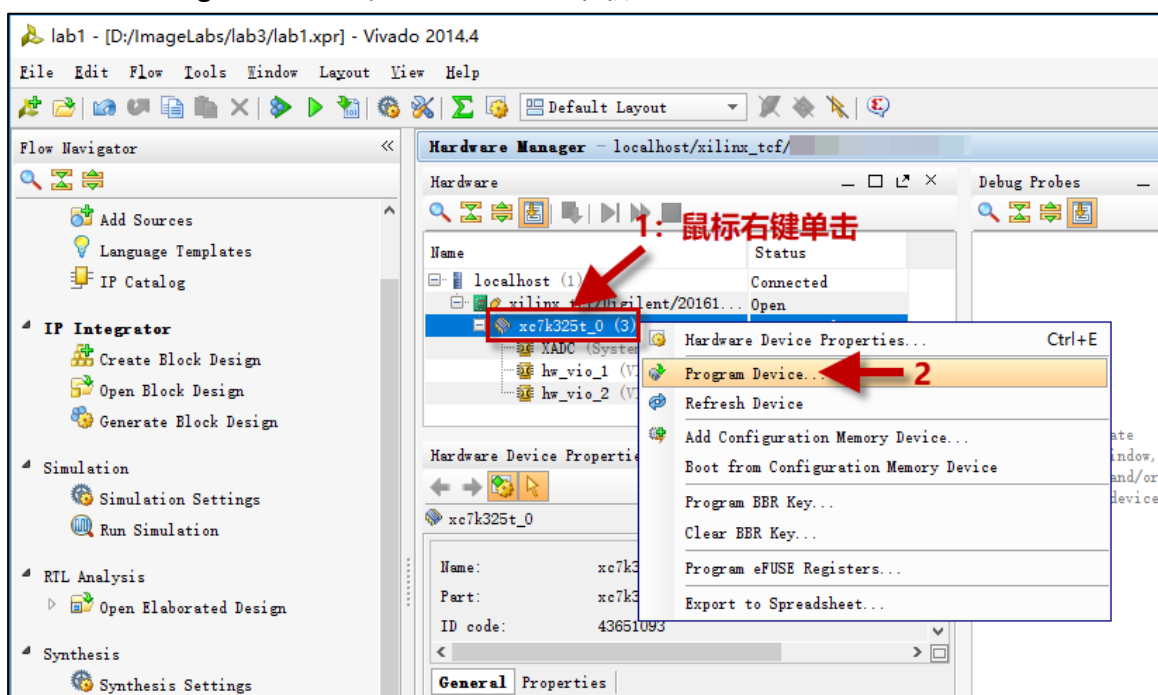



图 2-29 Program Device

在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	25 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

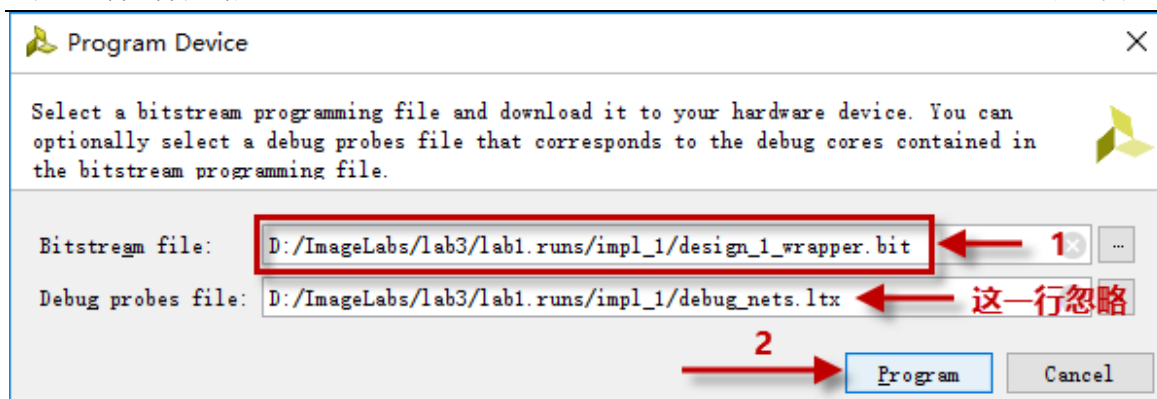


图 2-30 烧写目标器件

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

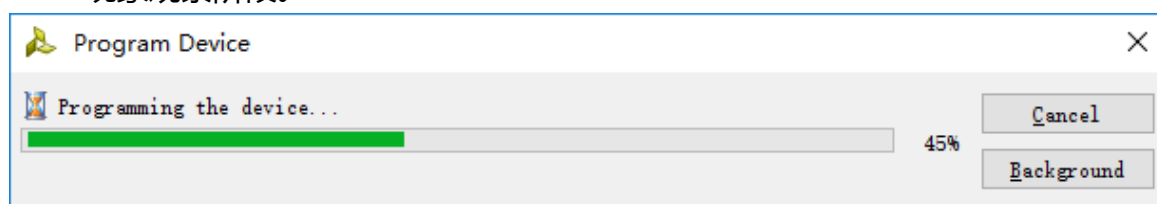



图 2-31 编程进度条

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	26 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

### 3. PARTA：对比度变换实验结果

首先我们让 HDMI 信号源显示一幅素材图片( 位于同文件夹下的 J20.png ),接着我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次，以便让信号源设备重新检测 ( Detect ) 一下接收设备，一切正常的话，我们即可在 HDMI 显示器上看到显示画面。

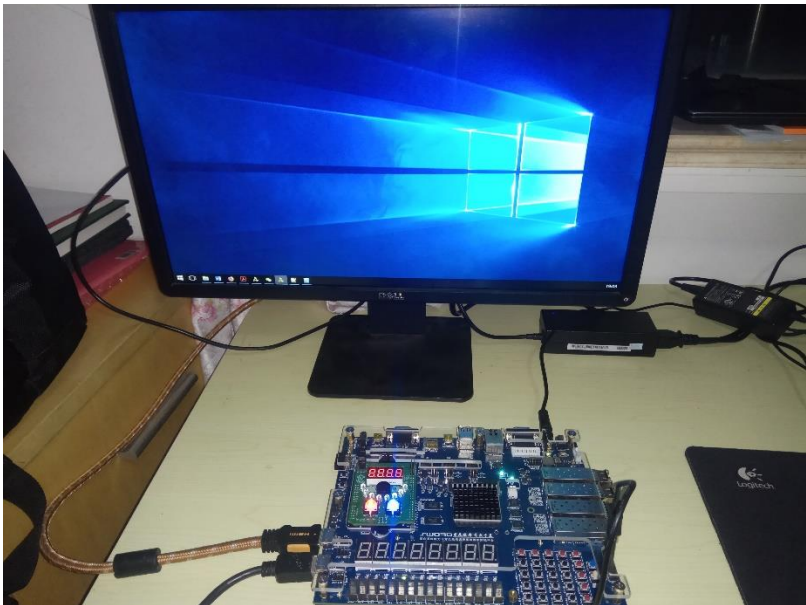


图 3-1 对比度变换显示结果

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	27 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

## 4. PARTB: 亮度变换实验流程

### 3.1 操作步骤

1. 完成了对比度变换部分的实验后，我们继续进行亮度变换部分的实验，回到 Vivado 的主界面，点击 Open Block Design 图标，然后在打开的 IP 设计视图的左边栏，点击 Add IP 的图标，过程如下图所示：

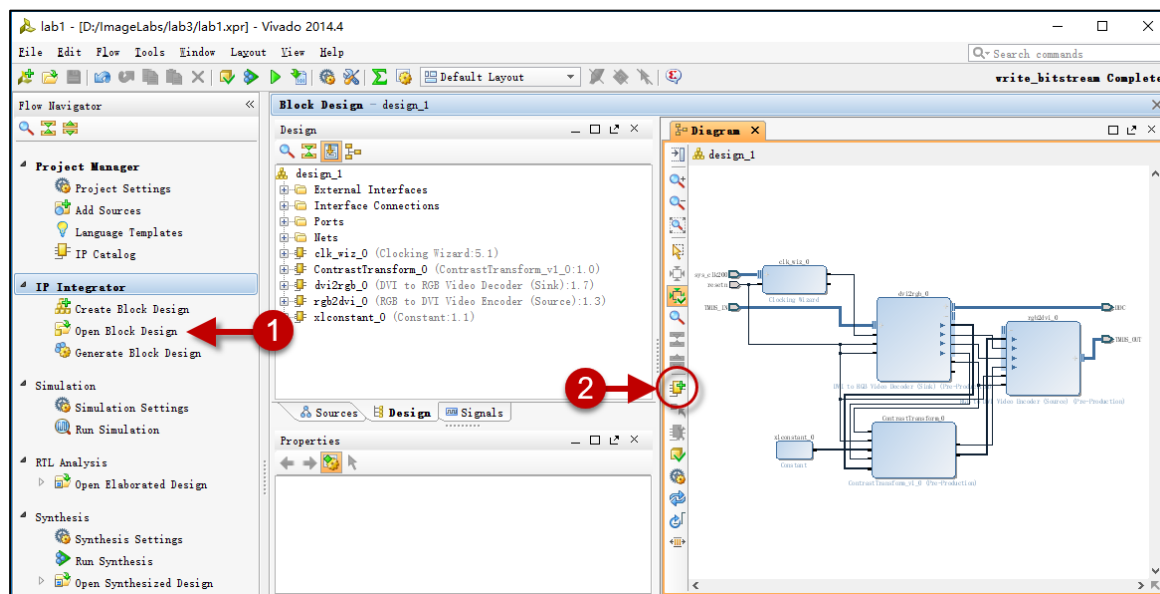


图 4-1 回到 Vivado 主界面继续添加 IP

在搜索栏中输入 Light，在搜索结果中双击 LightnessTransform\_v1\_0，添加亮度变换 IP，过程如下图所示：

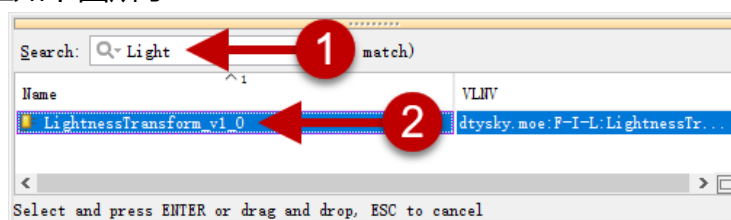



图 4-2 添加亮度变换 IP: LightnessTransform

2. 添加 LightnessTransform IP 后，和前面部分的实验一样，为了将这个 IP 接入数据流，我们需要断开之前的图像数据流，在此，我们只需删除对比度变换 IP，具体做法为：鼠标左键选中 ContrastTransform IP，当该 IP 为浅黄色之后，鼠标右键单击，在弹出的菜单中选择 Delete，过程如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	28 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

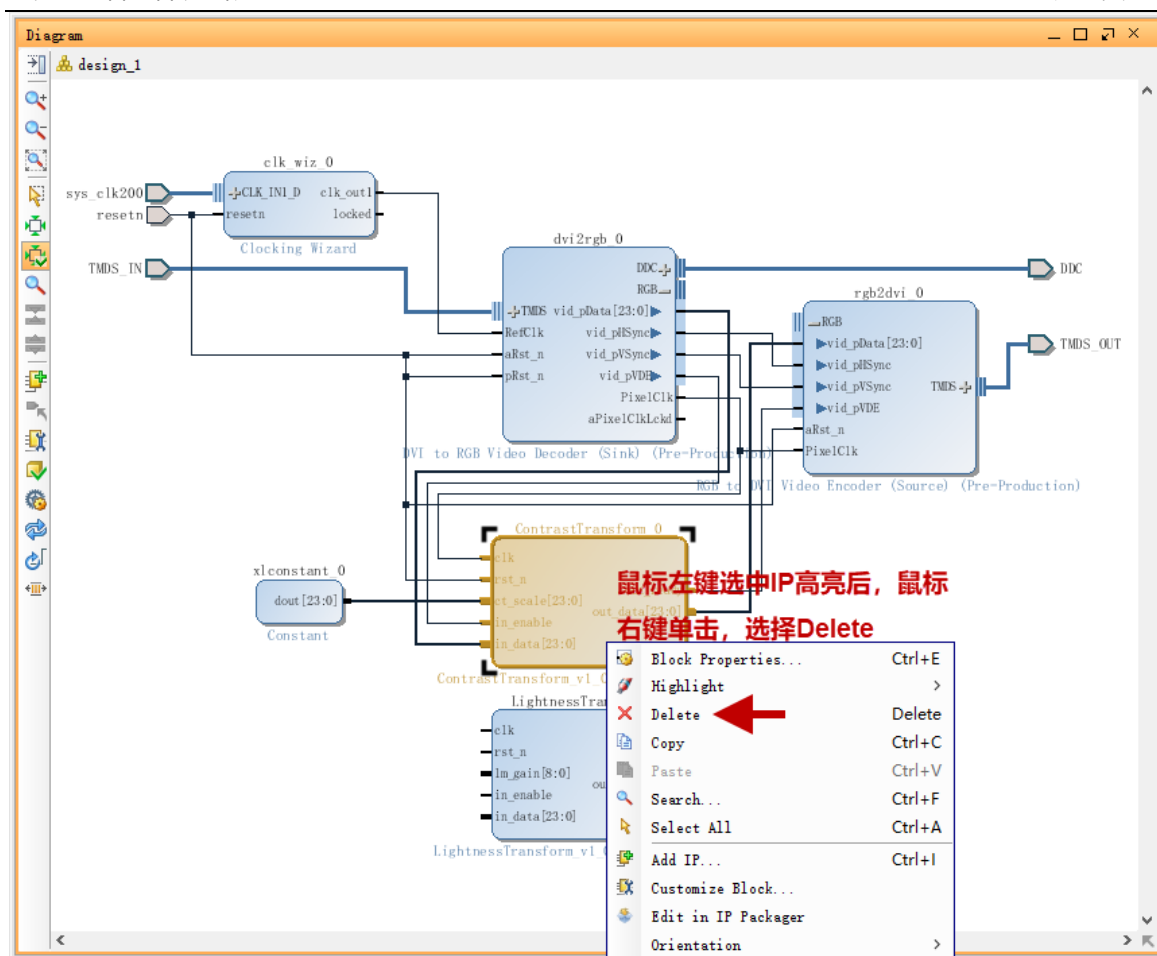


图 4-3 断开 RGB2DVI IP 的 vid\_pVDE 端口

3. 删除 ContrastTransform IP 后的效果如下图所示：

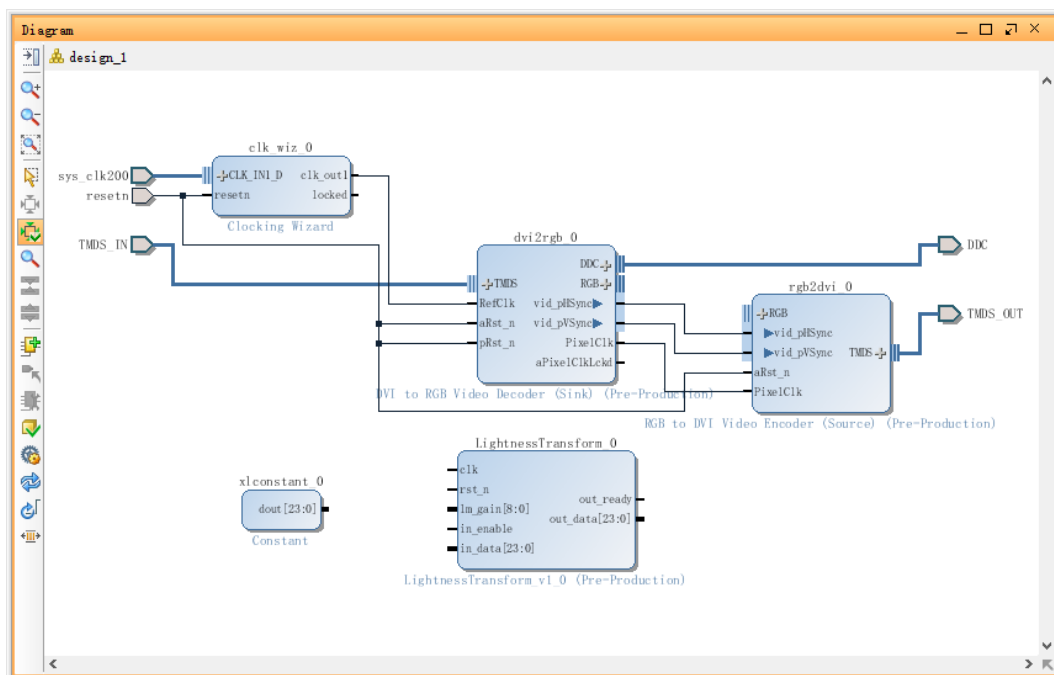



图 4-4 删除 IP 后的视图

4. 双击 LightnessTransform IP 进行配置, 在配置窗口中 将 Work Mode 设置为 Pipeline ,

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	29 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		



Color Channels 设置为 3，Color Width 设置为 8，点击 OK 确认，如下图所示：

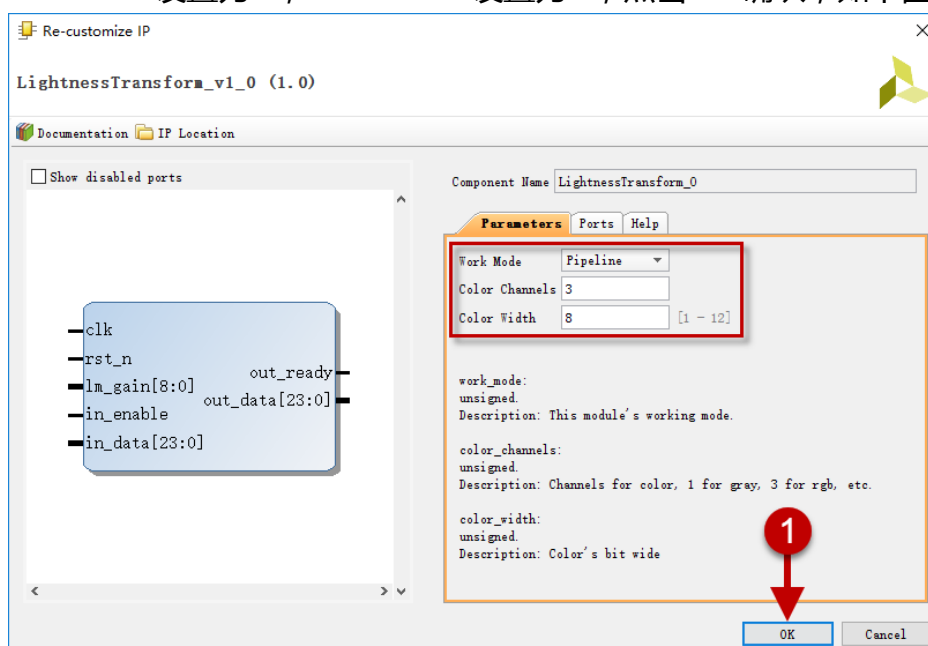


图 4-5 LightnessTransform IP 配置

5. 双击 xlconstant\_0 IP 进行配置，在配置窗口中，将 Const Width 改为 9，Const Val 设置为 90，完成设置后点 OK 继续，如下图所示：

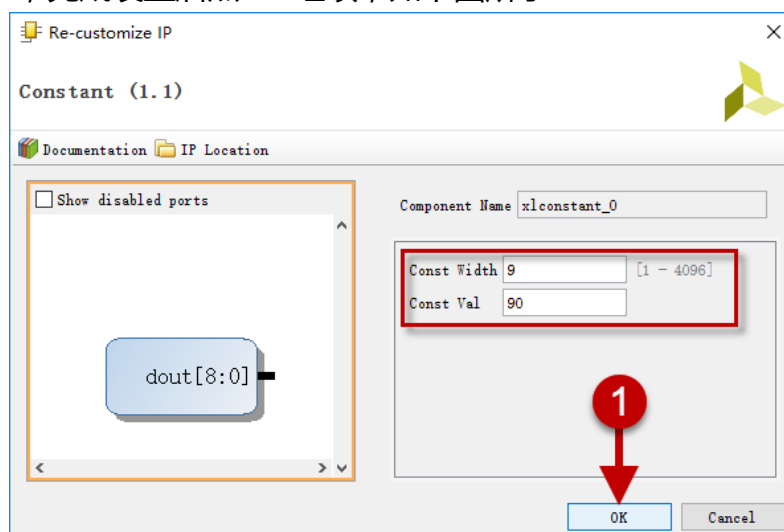



图 4-6 Constant IP 设置

6. 完成对 LightnessTransform 和 xlconstant 这两个 IP 的配置后，进行 IP 的连接，连接后的效果如下图所示，请仔细检查各 IP 的端口连接是否正确，为了方便核对，下图各种连接的高亮色图以示区别：

**提示：下图仅作为检查连接使用，读者完全不必也按照图示颜色进行标注 !!!**

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	30 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

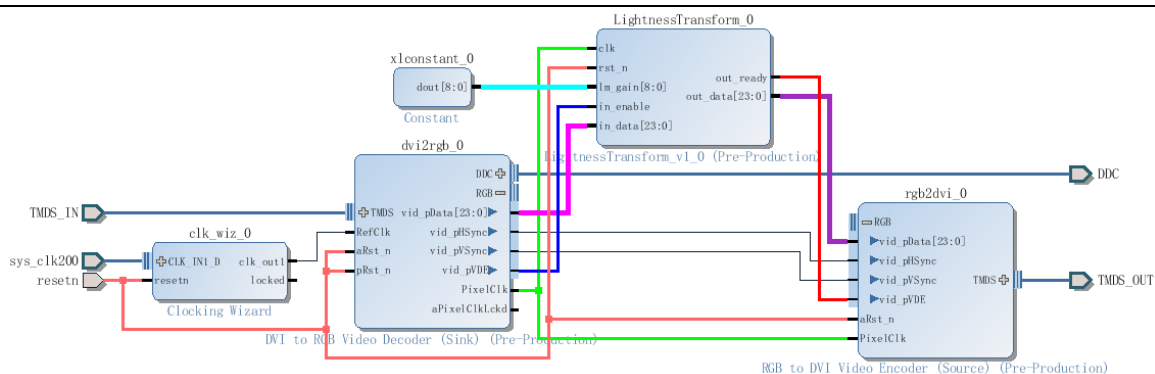


图 4-7 端口连接检查

7. 连接检查无误后，即可保存 IP 模块化设计，在 Vivado 主界面点击保存图标，如下图所示：

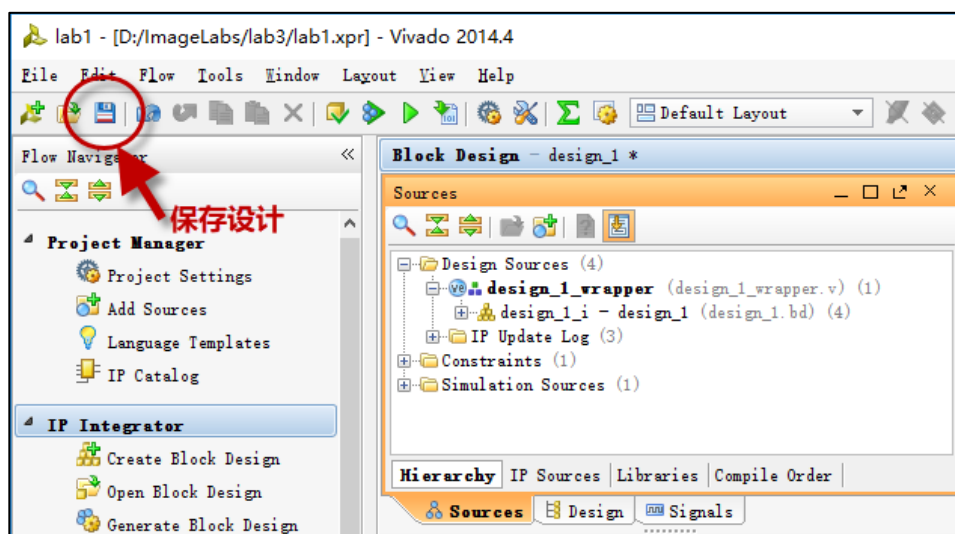


图 4-8 保存设计

8. 接着在 Source 子窗口中展开 design\_1\_wrapper，选中 design\_1.bd，鼠标右键单击，在弹出的菜单中选择 Create HDL Wrapper，整个过程如下图所示：

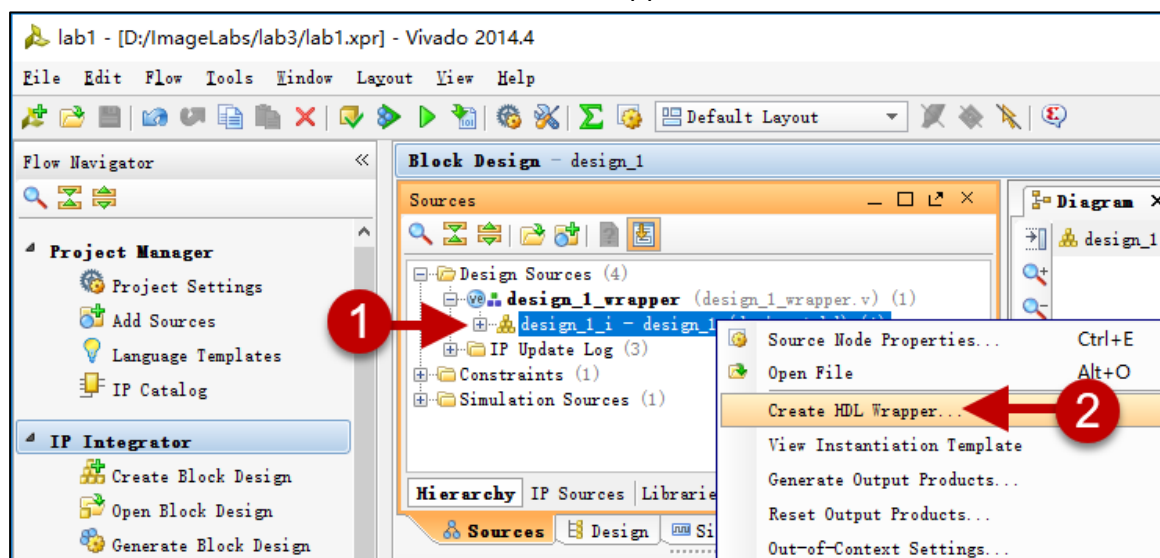



图 4-9 创建顶层文件

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	31 of 37
	作者	修改日期	公开	
Joseph Xu	2018/4/17			



9. 接着在弹出的对话框中,保持默认选项不变,即选择 Let Vivado manage wrapper and auto-update, 然后点击 OK, 如下图所示:

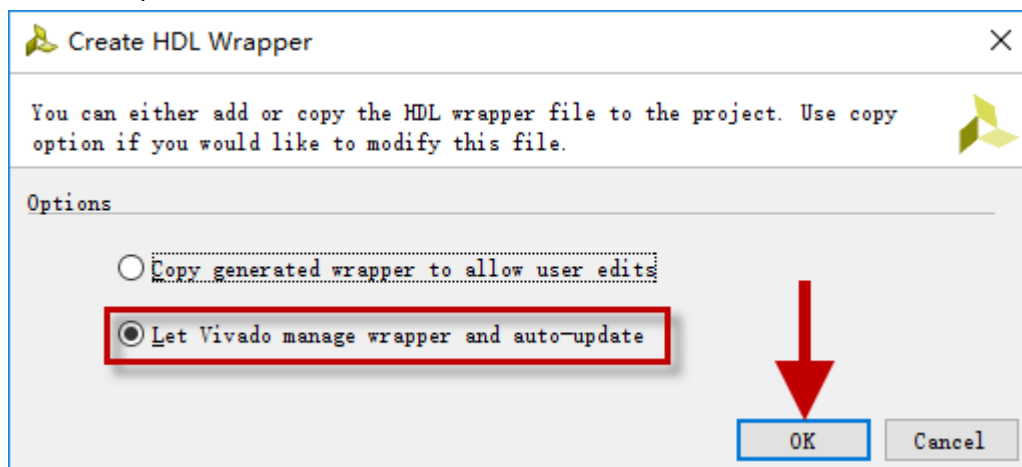


图 4-10 自动更新顶层文件

10. 在 Vivado 主界面点击 Generate Bitstream, 生成 bit 文件, 如下图所示:

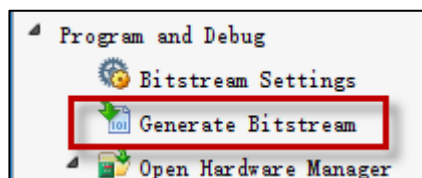


图 4-11 Generate Bitstream

在弹出的提示框中直接点 Yes 确认并继续, 如下图所示:

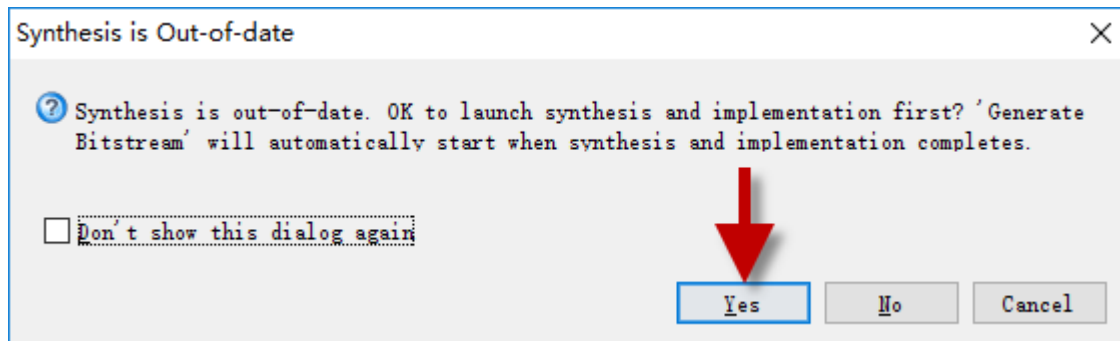



图 4-12 点击 Yes 确认生成 bit 文件

大约经过 10 分钟后, Vivado 会弹出 Bitstream Generation Completed 的提示框, 表示 bit 文件完成, 选择 Open Hardware Manager, 然后点击 OK, 如下图所示:

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	32 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

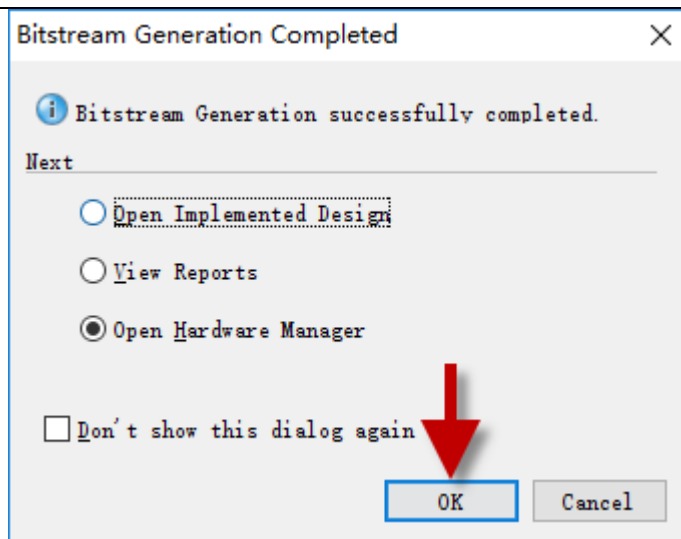



图 4-13 打开 Hardware Manager

11. 接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连接到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	33 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

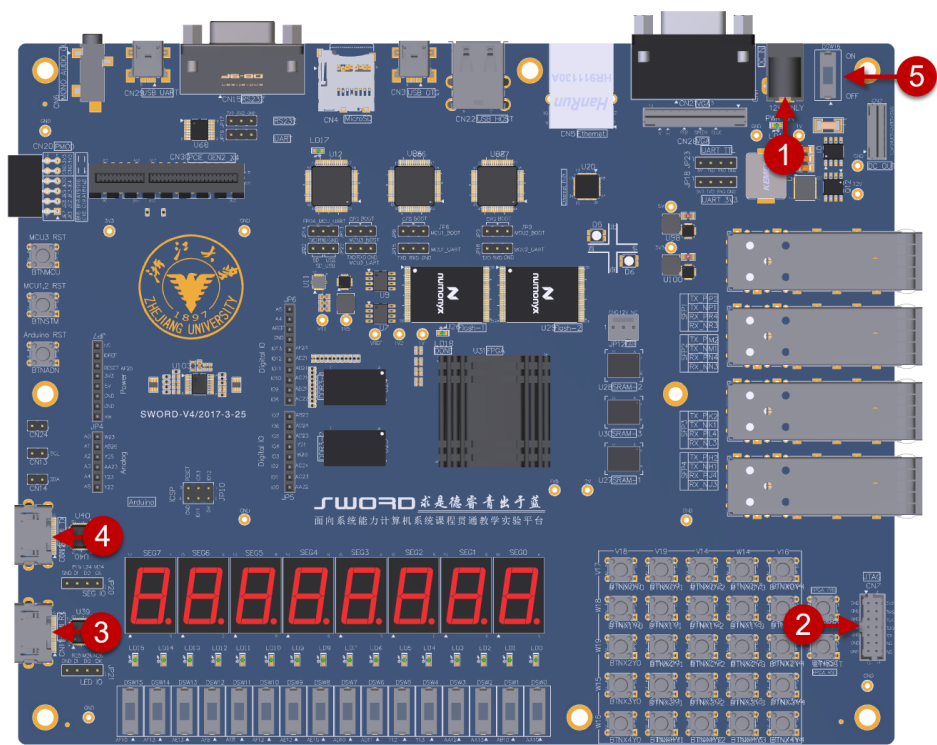


图 4-14 硬件连接对应位置

连接好后的效果如下图所示：

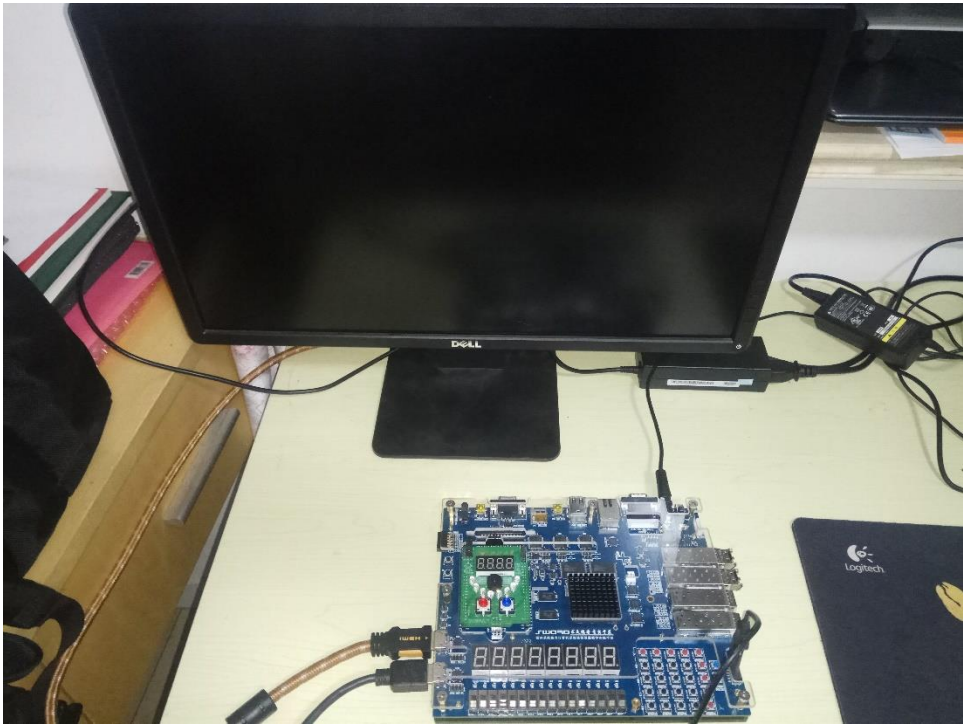


图 4-15 实际硬件连接

接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	34 of 37
	作者	修改日期	公开	
Joseph Xu		2018/4/17		

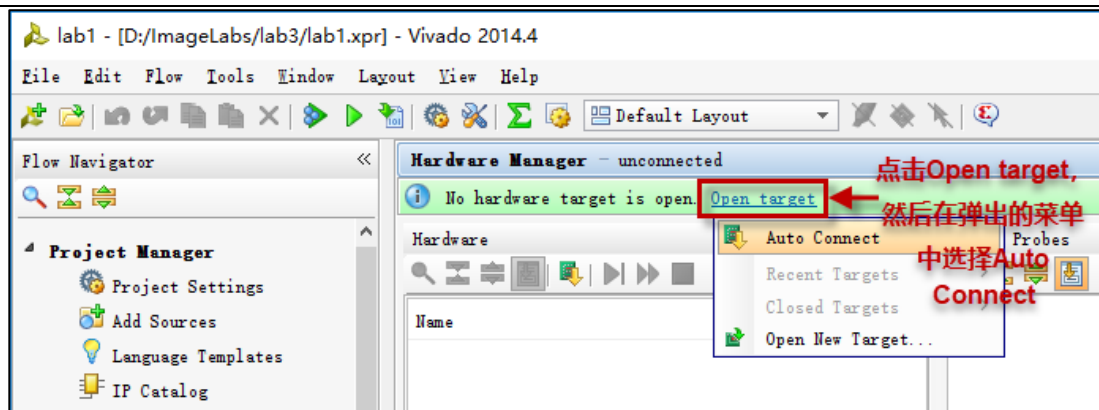


图 4-16 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

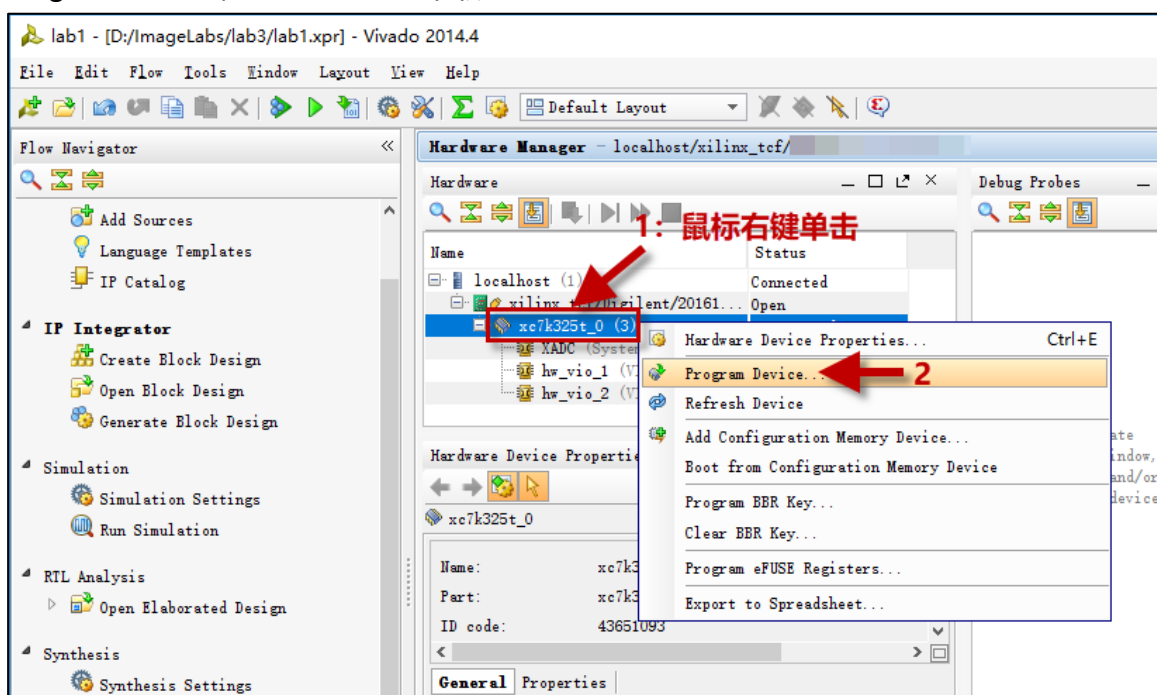


图 4-17 Program Device

在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	35 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

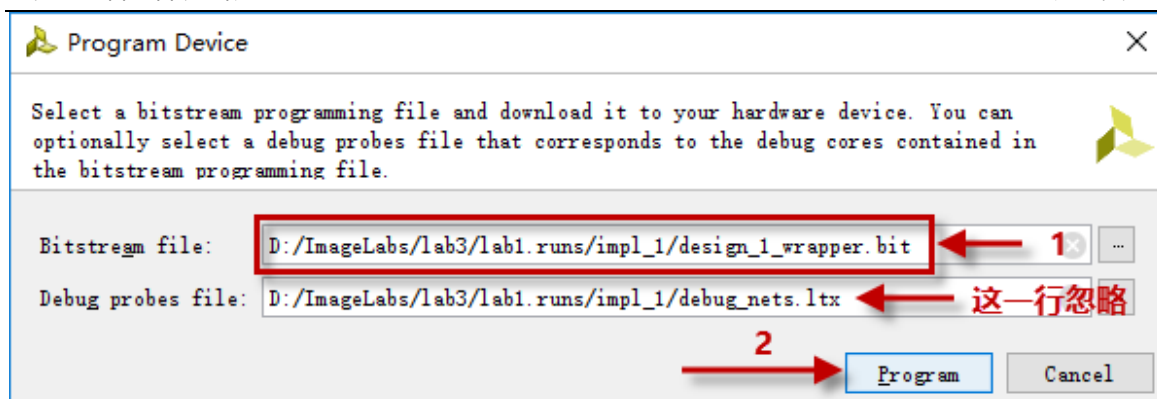


图 4-18 烧写目标器件

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

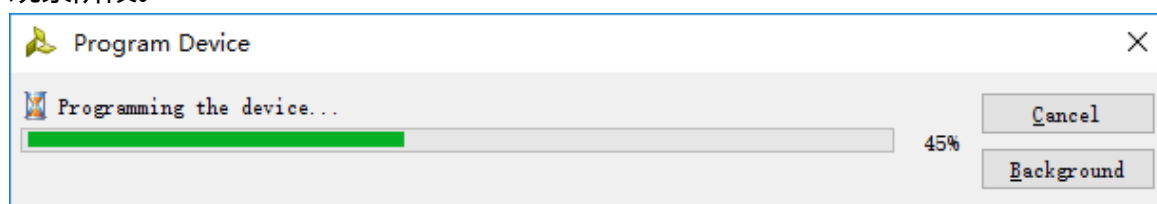



图 4-19 编程进度条

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	36 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		

5. PARTB 亮度变换实验结果

首先我们让 HDMI 信号源显示一幅素材图片( 位于同文件夹下的 J20.png ),接着我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次 , 以便让信号源设备重新检测 ( Detect ) 一下接收设备 , 一切正常的话 , 我们即可在 HDMI 显示器上看到显示画面。



图 5-1 亮度变换显示结果

	标题	文档编号	版本	页
	Lab3: 像素处理实验 2	XDI-SWORD-IMG-003	1.0	37 of 37
	作者	修改日期	公开	
	Joseph Xu	2018/4/17		